To our customers,

# Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

RENESAS

# 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.

User's Manual

32

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

# SH7137 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family

| | |
|---|---|
| SH7131 | R5F7131 |
| SH7132 | R5F7132 |
| SH7136 | R5F7136 |
| SH7137 | R5F7137 |

Renesas Electronics
www.renesas.com

Rev.3.00 2010.01

## Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com )
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

# General Precautions 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
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

The SH7131, SH7132, SH7136, and SH7137 Group RISC (Reduced Instruction Set Computer) microcomputers include a Renesas Technology-original RISC CPU as its core, and the peripheral functions required to configure a system.

Target Users: This manual was written for users who will be using the SH7131, SH7132, SH7136, and SH7137 Group 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 the SH7131, SH7132, SH7136, and SH7137 Group to the target users.
Refer to the SH-1/SH-2/SH-DSP Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip

  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions

  Read the SH-1/SH-2/SH-DSP 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 25, List of Registers.

| Examples: | | |
|---|---|---|
| | Register name: | The following notation is used for cases when the same or a similar function, e.g. serial communication interface, is implemented on more than one channel:<br>XXX_N (XXX is the register name and N is the channel number) |
| | Bit order: | The MSB is on the left and the LSB is on the right. |
| | Number notation: | Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx. |
| | Signal notation: | An overbar is added to a low-active signal: $\overline{\text{xxxx}}$ |

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require. http://www.renesas.com/

SH7131, SH7132, SH7136, and SH7137 Group manuals:

| Document Title | Document No. |
|---|---|
| SH7137 Group Hardware Manual | This manual |
| SH-1/SH-2/SH-DSP Software Manual | REJ09B0171 |

User's manuals for development tools:

| Document Title | Document No. |
|---|---|
| SuperH™ RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package V.9.00 User's Manual | REJ10B0152 |
| SuperH™ RISC engine High-performance Embedded Workshop 3 User's Manual | REJ10B0025 |
| SuperH RISC engine High-performance Embedded Workshop 3 Tutorial | REJ10B0023 |

Application note:

| Document Title | Document No. |
|---|---|
| SuperH RISC engine C/C++ Compiler Package Application Note | REJ05B0463 |

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

# Contents


Section 1 Overview ..... 1
1.1 Features of SH7131, SH7132, SH7136, and SH7137 ..... 1
1.2 Block Diagram ..... 7
1.3 Pin Assignments ..... 8
1.4 Pin Functions ..... 10

Section 2 CPU ..... 17
2.1 Features ..... 17
2.2 Register Configuration ..... 18
2.2.1 General Registers (Rn) ..... 19
2.2.2 Control Registers ..... 19
2.2.3 System Registers ..... 21
2.2.4 Initial Values of Registers ..... 21
2.3 Data Formats ..... 22
2.3.1 Register Data Format ..... 22
2.3.2 Memory Data Formats ..... 22
2.3.3 Immediate Data Formats ..... 23
2.4 Features of Instructions ..... 23
2.4.1 RISC Type ..... 23
2.4.2 Addressing Modes ..... 26
2.4.3 Instruction Formats ..... 29
2.5 Instruction Set ..... 33
2.5.1 Instruction Set by Type ..... 33
2.5.2 Data Transfer Instructions ..... 37
2.5.3 Arithmetic Operation Instructions ..... 39
2.5.4 Logic Operation Instructions ..... 41
2.5.5 Shift Instructions ..... 42
2.5.6 Branch Instructions ..... 43
2.5.7 System Control Instructions ..... 44
2.6 Processing States ..... 46

Section 3 MCU Operating Modes ..... 49
3.1 Selection of Operating Modes ..... 49
3.2 Input/Output Pins ..... 50
3.3 Operating Modes ..... 51
3.3.1 Mode 0 (MCU Extension Mode 0) ..... 51
3.3.2 Mode 2 (MCU Extension Mode 2) ..... 51

3.3.3 Mode 3 (Single Chip Mode) ..... 51
3.4 Address Map ..... 52
3.5 Initial State in This LSI ..... 56
3.6 Note on Changing Operating Mode ..... 56

Section 4 Clock Pulse Generator (CPG) ..... 57
4.1 Features ..... 57
4.2 Input/Output Pins ..... 61
4.3 Clock Operating Mode ..... 62
4.4 Register Descriptions ..... 67
4.4.1 Frequency Control Register (FRQCR) ..... 67
4.4.2 Oscillation Stop Detection Control Register (OSCCR) ..... 70
4.5 Changing Frequency ..... 71
4.6 Oscillator ..... 72
4.6.1 Connecting Crystal Resonator ..... 72
4.6.2 External Clock Input Method ..... 73
4.7 Function for Detecting Oscillator Stop ..... 74
4.8 Usage Notes ..... 75
4.8.1 Note on Crystal Resonator ..... 75
4.8.2 Notes on Board Design ..... 75

Section 5 Exception Handling ..... 77
5.1 Overview ..... 77
5.1.1 Types of Exception Handling and Priority ..... 77
5.1.2 Exception Handling Operations ..... 78
5.1.3 Exception Handling Vector Table ..... 79
5.2 Resets ..... 81
5.2.1 Types of Resets ..... 81
5.2.2 Power-On Reset ..... 81
5.2.3 Manual Reset ..... 82
5.3 Address Errors ..... 83
5.3.1 Address Error Sources ..... 83
5.3.2 Address Error Exception Source ..... 84
5.4 Interrupts ..... 85
5.4.1 Interrupt Sources ..... 85
5.4.2 Interrupt Priority ..... 86
5.4.3 Interrupt Exception Handling ..... 86
5.5 Exceptions Triggered by Instructions ..... 87
5.5.1 Types of Exceptions Triggered by Instructions ..... 87
5.5.2 Trap Instructions ..... 87

5.5.3 Illegal Slot Instructions ..... 88
5.5.4 General Illegal Instructions ..... 88
5.6 Cases when Exceptions are Accepted ..... 89
5.7 Stack States after Exception Handling Ends ..... 90
5.8 Usage Notes ..... 92
5.8.1 Value of Stack Pointer (SP) ..... 92
5.8.2 Value of Vector Base Register (VBR) ..... 92
5.8.3 Address Errors Caused by Stacking for Address Error Exception Handling ..... 92
5.8.4 Notes on Slot Illegal Instruction Exception Handling ..... 93

Section 6 Interrupt Controller (INTC) ..... 95
6.1 Features ..... 95
6.2 Input/Output Pins ..... 97
6.3 Register Descriptions ..... 98
6.3.1 Interrupt Control Register 0 (ICR0) ..... 99
6.3.2 IRQ Control Register (IRQCR) ..... 100
6.3.3 IRQ Status register (IRQSR) ..... 102
6.3.4 Interrupt Priority Registers A, D to F, and H to M (IPRA, IPRD to IPRF, and IPRH to IPRM) ..... 105
6.4 Interrupt Sources ..... 108
6.4.1 External Interrupts ..... 108
6.4.2 On-Chip Peripheral Module Interrupts ..... 109
6.4.3 User Break Interrupt (SH7136 and SH7137 only) ..... 109
6.5 Interrupt Exception Handling Vector Table ..... 110
6.6 Interrupt Operation ..... 114
6.6.1 Interrupt Sequence ..... 114
6.6.2 Stack after Interrupt Exception Handling ..... 117
6.7 Interrupt Response Time ..... 117
6.8 Data Transfer with Interrupt Request Signals ..... 119
6.8.1 Handling Interrupt Request Signals as Sources for DTC Activation and CPU Interrupts ..... 120
6.8.2 Handling Interrupt Request Signals as Sources for DTC Activation, but Not CPU Interrupts ..... 120
6.8.3 Handling Interrupt Request Signals as Sources for CPU Interrupts, but Not DTC Activation ..... 121
6.9 Usage Note ..... 121

Section 7 User Break Controller (UBC) (SH7136 and SH7137 only) ..... 123
7.1 Features ..... 123
7.2 Input/Output Pins ..... 125

7.3 Register Descriptions ..... 126
7.3.1 Break Address Register A (BARA) ..... 127
7.3.2 Break Address Mask Register A (BAMRA) ..... 127
7.3.3 Break Bus Cycle Register A (BBRA) ..... 128
7.3.4 Break Data Register A (BDRA) ..... 130
7.3.5 Break Data Mask Register A (BDMRA) ..... 131
7.3.6 Break Address Register B (BARB) ..... 132
7.3.7 Break Address Mask Register B (BAMRB) ..... 133
7.3.8 Break Data Register B (BDRB) ..... 134
7.3.9 Break Data Mask Register B (BDMRB) ..... 135
7.3.10 Break Bus Cycle Register B (BBRB) ..... 136
7.3.11 Break Control Register (BRCR) ..... 138
7.3.12 Execution Times Break Register (BETR) ..... 143
7.3.13 Branch Source Register (BRSR) ..... 144
7.3.14 Branch Destination Register (BRDR) ..... 145
7.4 Operation ..... 146
7.4.1 Flow of the User Break Operation ..... 146
7.4.2 User Break on Instruction Fetch Cycle ..... 147
7.4.3 Break on Data Access Cycle ..... 148
7.4.4 Sequential Break ..... 149
7.4.5 Value of Saved Program Counter ..... 149
7.4.6 PC Trace ..... 150
7.4.7 Usage Examples ..... 151
7.5 Usage Notes ..... 156

Section 8 Data Transfer Controller (DTC) ..... 159
8.1 Features ..... 159
8.2 Register Descriptions ..... 161
8.2.1 DTC Mode Register A (MRA) ..... 162
8.2.2 DTC Mode Register B (MRB) ..... 163
8.2.3 DTC Source Address Register (SAR) ..... 165
8.2.4 DTC Destination Address Register (DAR) ..... 165
8.2.5 DTC Transfer Count Register A (CRA) ..... 166
8.2.6 DTC Transfer Count Register B (CRB) ..... 167
8.2.7 DTC Enable Registers A to E (DTCERA to DTCERE) ..... 168
8.2.8 DTC Control Register (DTCCR) ..... 169
8.2.9 DTC Vector Base Register (DTCVBR) ..... 171
8.2.10 Bus Function Extending Register (BSCEHR) ..... 171
8.3 Activation Sources ..... 172
8.4 Location of Transfer Information and DTC Vector Table ..... 172

8.5 Operation ..... 177
8.5.1 Transfer Information Read Skip Function ..... 182
8.5.2 Transfer Information Writeback Skip Function ..... 183
8.5.3 Normal Transfer Mode ..... 183
8.5.4 Repeat Transfer Mode ..... 184
8.5.5 Block Transfer Mode ..... 186
8.5.6 Chain Transfer ..... 187
8.5.7 Operation Timing ..... 189
8.5.8 Number of DTC Execution Cycles ..... 192
8.5.9 DTC Bus Release Timing ..... 194
8.5.10 DTC Activation Priority Order ..... 197
8.6 DTC Activation by Interrupt ..... 198
8.7 Examples of Use of the DTC ..... 199
8.7.1 Normal Transfer Mode ..... 199
8.7.2 Chain Transfer when Counter = 0 ..... 199
8.8 Interrupt Sources ..... 201
8.9 Usage Notes ..... 201
8.9.1 Module Standby Mode Setting ..... 201
8.9.2 On-Chip RAM ..... 201
8.9.3 DTCE Bit Setting ..... 201
8.9.4 Chain Transfer ..... 201
8.9.5 Transfer Information Start Address, Source Address, and Destination Address ..... 201
8.9.6 Access to DTC Registers through DTC ..... 202
8.9.7 Notes on IRQ Interrupt as DTC Activation Source ..... 202
8.9.8 Notes on SCI as DTC Activation Sources ..... 202
8.9.9 Clearing Interrupt Source Flag ..... 202
8.9.10 Conflict between NMI Interrupt and DTC Activation ..... 202
8.9.11 Operation When a DTC Activation Request is Cancelled While in Progress ..... 202

Section 9 Bus State Controller (BSC) ..... 203
9.1 Features ..... 203
9.2 Input/Output Pins ..... 205
9.3 Area Overview ..... 205
9.3.1 Area Division ..... 205
9.3.2 Address Map ..... 205
9.4 Register Descriptions ..... 211
9.4.1 Common Control Register (CMNCR) ..... 211
9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 and 1) ..... 213
9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 and 1) ..... 216

9.4.4 Bus Function Extending Register (BSCEHR) ..... 218
9.5 Operation ..... 222
9.5.1 Endian/Access Size and Data Alignment ..... 222
9.5.2 Normal Space Interface ..... 223
9.5.3 Access Wait Control ..... 226
9.5.4 $\overline{\text{CSn}}$ Assert Period Extension ..... 228
9.5.5 Wait between Access Cycles ..... 229
9.5.6 Bus Arbitration ..... 232
9.5.7 Others ..... 236
9.5.8 Access to On-Chip FLASH and On-Chip RAM by CPU ..... 237
9.5.9 Access to On-Chip Peripheral I/O Registers by CPU ..... 237
9.5.10 Access to External Memory by CPU ..... 239

Section 10 Multi-Function Timer Pulse Unit 2 (MTU2) ..... 241
10.1 Features ..... 241
10.2 Input/Output Pins ..... 247
10.3 Register Descriptions ..... 248
10.3.1 Timer Control Register (TCR) ..... 252
10.3.2 Timer Mode Register (TMDR) ..... 256
10.3.3 Timer I/O Control Register (TIOR) ..... 259
10.3.4 Timer Compare Match Clear Register (TCNTCMPCLR) ..... 278
10.3.5 Timer Interrupt Enable Register (TIER) ..... 279
10.3.6 Timer Status Register (TSR) ..... 284
10.3.7 Timer Buffer Operation Transfer Mode Register (TBTM) ..... 292
10.3.8 Timer Input Capture Control Register (TICCR) ..... 293
10.3.9 Timer Synchronous Clear Register (TSYCR) ..... 295
10.3.10 Timer A/D Converter Start Request Control Register (TADCR) ..... 297
10.3.11 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4) ..... 300
10.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4) ..... 300
10.3.13 Timer Counter (TCNT) ..... 301
10.3.14 Timer General Register (TGR) ..... 301
10.3.15 Timer Start Register (TSTR) ..... 302
10.3.16 Timer Synchronous Register (TSYR) ..... 304
10.3.17 Timer Counter Synchronous Start Register (TCSYSTR) ..... 306
10.3.18 Timer Read/Write Enable Register (TRWER) ..... 309
10.3.19 Timer Output Master Enable Register (TOER) ..... 310
10.3.20 Timer Output Control Register 1 (TOCR1) ..... 311
10.3.21 Timer Output Control Register 2 (TOCR2) ..... 314

10.3.22 Timer Output Level Buffer Register (TOLBR) ..... 317
10.3.23 Timer Gate Control Register (TGCR) ..... 318
10.3.24 Timer Subcounter (TCNTS) ..... 320
10.3.25 Timer Dead Time Data Register (TDDR) ..... 321
10.3.26 Timer Cycle Data Register (TCDR) ..... 321
10.3.27 Timer Cycle Buffer Register (TCBR) ..... 322
10.3.28 Timer Interrupt Skipping Set Register (TITCR) ..... 322
10.3.29 Timer Interrupt Skipping Counter (TITCNT) ..... 324
10.3.30 Timer Buffer Transfer Set Register (TBTER) ..... 325
10.3.31 Timer Dead Time Enable Register (TDER) ..... 326
10.3.32 Timer Waveform Control Register (TWCR) ..... 327
10.3.33 Bus Master Interface ..... 329
10.4 Operation ..... 330
10.4.1 Basic Functions ..... 330
10.4.2 Synchronous Operation ..... 336
10.4.3 Buffer Operation ..... 338
10.4.4 Cascaded Operation ..... 342
10.4.5 PWM Modes ..... 347
10.4.6 Phase Counting Mode ..... 352
10.4.7 Reset-Synchronized PWM Mode ..... 359
10.4.8 Complementary PWM Mode ..... 362
10.4.9 A/D Converter Start Request Delaying Function ..... 406
10.4.10 MTU2–MTU2S Synchronous Operation ..... 410
10.4.11 External Pulse Width Measurement ..... 416
10.4.12 Dead Time Compensation ..... 417
10.4.13 TCNT Capture at Crest and/or Trough in Complementary PWM Operation ..... 419
10.5 Interrupt Sources ..... 420
10.5.1 Interrupt Sources and Priorities ..... 420
10.5.2 DTC Activation ..... 422
10.5.3 A/D Converter Activation ..... 423
10.6 Operation Timing ..... 425
10.6.1 Input/Output Timing ..... 425
10.6.2 Interrupt Signal Timing ..... 432
10.7 Usage Notes ..... 437
10.7.1 Module Standby Mode Setting ..... 437
10.7.2 Input Clock Restrictions ..... 437
10.7.3 Caution on Period Setting ..... 438
10.7.4 Contention between TCNT Write and Clear Operations ..... 438
10.7.5 Contention between TCNT Write and Increment Operations ..... 439
10.7.6 Contention between TGR Write and Compare Match ..... 440

10.7.7 Contention between Buffer Register Write and Compare Match ..... 441
10.7.8 Contention between Buffer Register Write and TCNT Clear ..... 442
10.7.9 Contention between TGR Read and Input Capture..... 443
10.7.10 Contention between TGR Write and Input Capture..... 444
10.7.11 Contention between Buffer Register Write and Input Capture ..... 445
10.7.12 TCNT_2 Write and Overflow/Underflow Contention in Cascade Connection .... 445
10.7.13 Counter Value during Complementary PWM Mode Stop ..... 447
10.7.14 Buffer Operation Setting in Complementary PWM Mode ..... 447
10.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag ..... 448
10.7.16 Overflow Flags in Reset Synchronous PWM Mode ..... 449
10.7.17 Contention between Overflow/Underflow and Counter Clearing..... 450
10.7.18 Contention between TCNT Write and Overflow/Underflow..... 451
10.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode ..... 451
10.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode ..... 452
10.7.21 Interrupts in Module Standby Mode ..... 452
10.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection..... 452
10.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode ..... 453
10.8 MTU2 Output Pin Initialization ..... 455
10.8.1 Operating Modes ..... 455
10.8.2 Reset Start Operation ..... 455
10.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc. ..... 456
10.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc. ..... 457

Section 11 Multi-Function Timer Pulse Unit 2S (MTU2S) .....487
11.1 Input/Output Pins ..... 491
11.2 Register Descriptions ..... 492

Section 12 Port Output Enable (POE) .....495
12.1 Features ..... 495
12.2 Input/Output Pins ..... 497
12.3 Register Descriptions ..... 499
12.3.1 Input Level Control/Status Register 1 (ICSR1) ..... 500
12.3.2 Output Level Control/Status Register 1 (OCSR1) ..... 503
12.3.3 Input Level Control/Status Register 2 (ICSR2) ..... 504
12.3.4 Output Level Control/Status Register 2 (OCSR2) ..... 507
12.3.5 Input Level Control/Status Register 3 (ICSR3) ..... 509

12.3.6 Software Port Output Enable Register (SPOER) ..... 511
12.3.7 Port Output Enable Control Register 1 (POECR1) ..... 512
12.3.8 Port Output Enable Control Register 2 (POECR2) ..... 514
12.4 Operation ..... 517
12.4.1 Input Level Detection Operation ..... 518
12.4.2 Output-Level Compare Operation ..... 519
12.4.3 Release from High-Impedance State ..... 520
12.5 Interrupts ..... 521
12.6 Usage Note ..... 521
12.6.1 Pin State when a Power-On Reset is Issued from the Watchdog Timer ..... 521

Section 13 Watchdog Timer (WDT) ..... 523
13.1 Features ..... 523
13.2 Input/Output Pin for WDT ..... 525
13.3 Register Descriptions ..... 526
13.3.1 Watchdog Timer Counter (WTCNT) ..... 526
13.3.2 Watchdog Timer Control/Status Register (WTCSR) ..... 527
13.3.3 Notes on Register Access ..... 529
13.4 Operation ..... 530
13.4.1 Revoking Software Standbys ..... 530
13.4.2 Using Watchdog Timer Mode ..... 530
13.4.3 Using Interval Timer Mode ..... 531
13.5 Interrupt Source ..... 532
13.6 Usage Note ..... 532
13.6.1 WTCNT Setting Value ..... 532

Section 14 Serial Communication Interface (SCI) ..... 533
14.1 Features ..... 533
14.2 Input/Output Pins ..... 535
14.3 Register Descriptions ..... 536
14.3.1 Receive Shift Register (SCRSR) ..... 537
14.3.2 Receive Data Register (SCRDR) ..... 537
14.3.3 Transmit Shift Register (SCTSR) ..... 537
14.3.4 Transmit Data Register (SCTDR) ..... 538
14.3.5 Serial Mode Register (SCSMR) ..... 538
14.3.6 Serial Control Register (SCSCR) ..... 541
14.3.7 Serial Status Register (SCSSR) ..... 544
14.3.8 Serial Port Register (SCSPTR) ..... 550
14.3.9 Serial Direction Control Register (SCSDCR) ..... 552
14.3.10 Bit Rate Register (SCBRR) ..... 553

14.4 Operation ..... 563
14.4.1 Overview ..... 563
14.4.2 Operation in Asynchronous Mode ..... 565
14.4.3 Clock Synchronous Mode ..... 575
14.4.4 Multiprocessor Communication Function ..... 584
14.4.5 Multiprocessor Serial Data Transmission ..... 586
14.4.6 Multiprocessor Serial Data Reception ..... 587
14.5 SCI Interrupt Sources and DTC ..... 590
14.6 Serial Port Register (SCSPTR) and SCI Pins ..... 591
14.7 Usage Notes ..... 593
14.7.1 SCTDR Writing and TDRE Flag ..... 593
14.7.2 Multiple Receive Error Occurrence ..... 593
14.7.3 Break Detection and Processing ..... 594
14.7.4 Sending a Break Signal ..... 594
14.7.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) ..... 594
14.7.6 Note on Using DTC ..... 596
14.7.7 Note on Using External Clock in Clock Synchronous Mode ..... 596
14.7.8 Module Standby Mode Setting ..... 596

Section 15 Synchronous Serial Communication Unit (SSU) ..... 597
15.1 Features ..... 597
15.2 Input/Output Pins ..... 599
15.3 Register Descriptions ..... 600
15.3.1 SS Control Register H (SSCRH) ..... 601
15.3.2 SS Control Register L (SSCRL) ..... 603
15.3.3 SS Mode Register (SSMR) ..... 604
15.3.4 SS Enable Register (SSER) ..... 606
15.3.5 SS Status Register (SSSR) ..... 607
15.3.6 SS Control Register 2 (SSCR2) ..... 610
15.3.7 SS Transmit Data Registers 0 to 3 (SSTDR0 to SSTDR3) ..... 611
15.3.8 SS Receive Data Registers 0 to 3 (SSRDR0 to SSRDR3) ..... 612
15.3.9 SS Shift Register (SSTRSR) ..... 613
15.4 Operation ..... 614
15.4.1 Transfer Clock ..... 614
15.4.2 Relationship of Clock Phase, Polarity, and Data ..... 614
15.4.3 Relationship between Data Input/Output Pins and Shift Register ..... 615
15.4.4 Communication Modes and Pin Functions ..... 617
15.4.5 SSU Mode ..... 619
15.4.6 $\overline{\text{SCS}}$ Pin Control and Conflict Error ..... 628
15.4.7 Clock Synchronous Communication Mode ..... 630

15.5 SSU Interrupt Sources and DTC ..... 636
15.6 Usage Notes ..... 637
15.6.1 Module Standby Mode Setting ..... 637
15.6.2 Access to SSTDR and SSRDR Registers ..... 637
15.6.3 Continuous Transmission/Reception in SSU Slave Mode ..... 637
15.6.4 Note for Reception Operations in SSU Slave Mode ..... 637
15.6.5 Note on Master Transmission and Master Reception Operations in SSU Mode ..... 638
15.6.6 Note on DTC Transfers ..... 638

Section 16 $I^2C$ Bus Interface 2 ($I^2C2$) ..... 639
16.1 Features ..... 639
16.2 Input/Output Pins ..... 642
16.3 Register Descriptions ..... 643
16.3.1 $I^2C$ Bus Control Register 1 (ICCR1) ..... 643
16.3.2 $I^2C$ Bus Control Register 2 (ICCR2) ..... 646
16.3.3 $I^2C$ Bus Mode Register (ICMR) ..... 648
16.3.4 $I^2C$ Bus Interrupt Enable Register (ICIER) ..... 650
16.3.5 $I^2C$ Bus Status Register (ICSR) ..... 652
16.3.6 $I^2C$ Bus Slave Address Register (SAR) ..... 655
16.3.7 $I^2C$ Bus Transmit Data Register (ICDRT) ..... 656
16.3.8 $I^2C$ Bus Receive Data Register (ICDRR) ..... 656
16.3.9 $I^2C$ Bus Shift Register (ICDRS) ..... 656
16.3.10 NF2CYC Register (NF2CYC) ..... 657
16.4 Operation ..... 658
16.4.1 $I^2C$ Bus Format ..... 658
16.4.2 Master Transmit Operation ..... 659
16.4.3 Master Receive Operation ..... 661
16.4.4 Slave Transmit Operation ..... 664
16.4.5 Slave Receive Operation ..... 667
16.4.6 Clock Synchronous Serial Format ..... 668
16.4.7 Noise Filter ..... 672
16.4.8 Example of Use ..... 673
16.5 $I^2C2$ Interrupt Sources ..... 677
16.6 Operation Using the DTC ..... 678
16.7 Bit Synchronous Circuit ..... 679
16.8 Usage Note ..... 681
16.8.1 Module Standby Mode Setting ..... 681
16.8.2 Issuance of Stop Condition and Repeated Start Condition ..... 681
16.8.3 Issuance of a Start Condition and Stop Condition in Sequence ..... 681
16.8.4 Settings for Multi-Master Operation ..... 682

16.8.5 Reading ICDRR in Master Receive Mode ..... 682
16.8.6 Supported Emulator ..... 682

Section 17 A/D Converter (ADC) ..... 683
17.1 Features ..... 683
17.2 Input/Output Pins ..... 686
17.3 Register Descriptions ..... 687
17.3.1 A/D Control Registers_0 and _1 (ADCR_0 and ADCR_1) ..... 688
17.3.2 A/D Status Registers_0 and _1 (ADSR_0 and ADSR_1) ..... 691
17.3.3 A/D Start Trigger Select Registers_0 and _1 (ADSTRGR_0 and ADSTRGR_1) ..... 692
17.3.4 A/D Analog Input Channel Select Registers_0 and _1 (ADANSR_0 and ADANSR_1) ..... 694
17.3.5 A/D Data Registers 0 to 15 (ADDR0 to ADDR15) ..... 695
17.3.6 CPU Interface ..... 696
17.4 Operation ..... 697
17.4.1 Single-Cycle Scan Mode ..... 697
17.4.2 Continuous Scan Mode ..... 699
17.4.3 Input Sampling and A/D Conversion Time ..... 701
17.4.4 A/D Converter Activation by MTU2 and MTU2S ..... 703
17.4.5 External Trigger Input Timing ..... 704
17.4.6 Example of ADDR Auto-Clear Function ..... 704
17.5 Interrupt Sources and DTC Transfer Requests ..... 706
17.6 Definitions of A/D Conversion Accuracy ..... 707
17.7 Usage Notes ..... 709
17.7.1 Analog Input Voltage Range ..... 709
17.7.2 Relationship between AVcc, AVss and Vcc, Vss ..... 709
17.7.3 Range of $AV_{refh}$ and $AV_{refl}$ Pin Settings ..... 709
17.7.4 Notes on Board Design ..... 709
17.7.5 Notes on Noise Countermeasures ..... 710
17.7.6 Notes on Register Setting ..... 710

Section 18 Compare Match Timer (CMT) ..... 711
18.1 Features ..... 711
18.2 Register Descriptions ..... 712
18.2.1 Compare Match Timer Start Register (CMSTR) ..... 713
18.2.2 Compare Match Timer Control/Status Register (CMCSR) ..... 713
18.2.3 Compare Match Counter (CMCNT) ..... 715
18.2.4 Compare Match Constant Register (CMCOR) ..... 715
18.3 Operation ..... 716

18.3.1 Interval Count Operation ..... 716
18.3.2 CMCNT Count Timing ..... 716
18.4 Interrupts ..... 717
18.4.1 CMT Interrupt Sources and DTC Activation ..... 717
18.4.2 Timing of Setting Compare Match Flag ..... 718
18.4.3 Timing of Clearing Compare Match Flag ..... 718
18.5 Usage Notes ..... 719
18.5.1 Module Standby Mode Setting ..... 719
18.5.2 Conflict between Write and Compare-Match Processes of CMCNT ..... 719
18.5.3 Conflict between Word-Write and Count-Up Processes of CMCNT ..... 720
18.5.4 Conflict between Byte-Write and Count-Up Processes of CMCNT ..... 721
18.5.5 Compare Match between CMCNT and CMCOR ..... 721

Section 19 Controller Area Network (RCAN-ET) ..... 723
19.1 Summary ..... 723
19.1.1 Overview ..... 723
19.1.2 Scope ..... 723
19.1.3 Audience ..... 723
19.1.4 References ..... 724
19.1.5 Features ..... 724
19.2 Architecture ..... 725
19.3 Programming Model – Overview ..... 728
19.3.1 Memory Map ..... 728
19.3.2 Mailbox Structure ..... 729
19.3.3 RCAN-ET Control Registers ..... 737
19.3.4 RCAN-ET Mailbox Registers ..... 756
19.4 Application Note ..... 766
19.4.1 Test Mode Settings ..... 766
19.4.2 Configuration of RCAN-ET ..... 767
19.4.3 Message Transmission Sequence ..... 773
19.4.4 Message Receive Sequence ..... 776
19.4.5 Reconfiguration of Mailbox ..... 778
19.5 Interrupt Sources ..... 780
19.6 DTC Interface ..... 781
19.7 CAN Bus Interface ..... 782
19.8 Usage Notes ..... 783
19.8.1 Module Stop Mode ..... 783
19.8.2 Reset ..... 783
19.8.3 CAN Sleep Mode ..... 783
19.8.4 Register Access ..... 783

19.8.5 Interrupts ..... 784

Section 20 Pin Function Controller (PFC) ..... 785
20.1 Register Descriptions ..... 803
20.1.1 Port A I/O Register L (PAIORL) ..... 804
20.1.2 Port A Control Registers L1 to L4 (PACRL1 to PACRL4) ..... 804
20.1.3 Port B I/O Register L (PBIORL) ..... 819
20.1.4 Port B Control Registers L1, L2 (PBCRL1, PBCRL2) ..... 819
20.1.5 Port D I/O Register L (PDIORL) (SH7132 and SH7137 only) ..... 827
20.1.6 Port D Control Registers L1 to L3 (PDCRL1 to PDCRL3) (SH7132 and SH7137 only) ..... 828
20.1.7 Port E I/O Registers L, H (PEIORL, PEIORH) ..... 833
20.1.8 Port E Control Registers L1 to L4, H1, H2 (PECRL1 to PECRL4, PECRH1, PECRH2) ..... 834
20.1.9 IRQOUT Function Control Register (IFCR) ..... 851
20.2 Usage Notes ..... 852

Section 21 I/O Ports ..... 853
21.1 Port A ..... 854
21.1.1 Register Descriptions ..... 856
21.1.2 Port A Data Register L (PADRL) ..... 856
21.1.3 Port A Port Register L (PAPRL) ..... 858
21.2 Port B ..... 859
21.2.1 Register Descriptions ..... 860
21.2.2 Port B Data Register L (PBDRL) ..... 860
21.2.3 Port B Port Register L (PBPRL) ..... 863
21.3 Port D (SH7132/SH7137 Only) ..... 865
21.3.1 Register Descriptions ..... 865
21.3.2 Port D Data Register L (PDDRL) ..... 866
21.3.3 Port D Port Register L (PDPRL) ..... 867
21.4 Port E ..... 869
21.4.1 Register Descriptions ..... 871
21.4.2 Port E Data Registers H and L (PEDRH and PEDRL) ..... 871
21.4.3 Port E Port Registers H and L (PEPRH and PEPRL) ..... 874
21.5 Port F ..... 876
21.5.1 Register Descriptions ..... 877
21.5.2 Port F Data Register L (PFDRL) ..... 878

Section 22 Flash Memory ..... 881
22.1 Features ..... 881

22.2 Overview ..... 883
22.2.1 Block Diagram ..... 883
22.2.2 Operating Mode ..... 884
22.2.3 Mode Comparison ..... 886
22.2.4 Flash Memory Configuration ..... 887
22.2.5 Block Division ..... 888
22.2.6 Programming/Erasing Interface ..... 888
22.3 Input/Output Pins ..... 891
22.4 Register Descriptions ..... 891
22.4.1 Registers ..... 891
22.4.2 Programming/Erasing Interface Registers ..... 894
22.4.3 Programming/Erasing Interface Parameters ..... 901
22.4.4 RAM Emulation Register (RAMER) ..... 916
22.5 On-Board Programming Mode ..... 918
22.5.1 Boot Mode ..... 918
22.5.2 User Program Mode ..... 922
22.5.3 User Boot Mode ..... 932
22.6 Protection ..... 937
22.6.1 Hardware Protection ..... 937
22.6.2 Software Protection ..... 938
22.6.3 Error Protection ..... 938
22.7 Flash Memory Emulation in RAM ..... 940
22.8 Usage Notes ..... 943
22.8.1 Switching between User MAT and User Boot MAT ..... 943
22.8.2 Interrupts during Programming/Erasing ..... 944
22.8.3 Other Notes ..... 947
22.9 Supplementary Information ..... 949
22.9.1 Specifications of the Standard Serial Communications Interface in Boot Mode ..... 949
22.9.2 Areas for Storage of the Procedural Program and Data for Programming ..... 979
22.10 Programmer Mode ..... 986

Section 23 RAM ..... 987
23.1 Usage Notes ..... 988
23.1.1 Module Standby Mode Setting ..... 988
23.1.2 Address Error ..... 988
23.1.3 Initial Values in RAM ..... 988

Section 24 Power-Down Modes ..... 989
24.1 Features ..... 989

24.1.1 Types of Power-Down Modes ..... 989
24.2 Input/Output Pins ..... 991
24.3 Register Descriptions ..... 992
24.3.1 Standby Control Register 1 (STBCR1) ..... 992
24.3.2 Standby Control Register 2 (STBCR2) ..... 993
24.3.3 Standby Control Register 3 (STBCR3) ..... 994
24.3.4 Standby Control Register 4 (STBCR4) ..... 996
24.3.5 Standby Control Register 5 (STBCR5) ..... 997
24.3.6 Standby Control Register 6 (STBCR6) ..... 998
24.3.7 RAM Control Register (RAMCR) ..... 999
24.4 Sleep Mode ..... 1000
24.4.1 Transition to Sleep Mode ..... 1000
24.4.2 Canceling Sleep Mode ..... 1000
24.5 Software Standby Mode (SH7136 and SH7137 only) ..... 1001
24.5.1 Transition to Software Standby Mode ..... 1001
24.5.2 Canceling Software Standby Mode ..... 1002
24.6 Deep Software Standby Mode (SH7136 and SH7137 only) ..... 1003
24.6.1 Transition to Deep Software Standby Mode ..... 1003
24.6.2 Canceling Deep Software Standby Mode ..... 1003
24.7 Module Standby Mode ..... 1004
24.7.1 Transition to Module Standby Mode ..... 1004
24.7.2 Canceling Module Standby Function ..... 1004
24.8 Usage Note ..... 1005
24.8.1 Current Consumption while Waiting for Oscillation to be Stabilized ..... 1005
24.8.2 Executing the SLEEP Instruction ..... 1005

Section 25 List of Registers ..... 1007
25.1 Register Address Table (In the Order of Addresses) ..... 1008
25.2 Register Bit List ..... 1029
25.3 Register States in Each Operating Mode ..... 1050

Section 26 Electrical Characteristics ..... 1063
26.1 Absolute Maximum Ratings ..... 1063
26.2 DC Characteristics ..... 1064
26.3 AC Characteristics ..... 1071
26.3.1 Clock Timing ..... 1072
26.3.2 Control Signal Timing ..... 1075
26.3.3 AC Bus Timing ..... 1078
26.3.4 Multi Function Timer Pulse Unit 2 (MTU2) Timing ..... 1085
26.3.5 Multi Function Timer Pulse Unit 2S (MTU2S) Timing ..... 1087

26.3.6 I/O Port Timing ..... 1088
26.3.7 Watchdog Timer (WDT) Timing ..... 1089
26.3.8 Serial Communication Interface (SCI) Timing ..... 1090
26.3.9 Synchronous Serial Communication Unit (SSU) Timing ..... 1092
26.3.10 Controller Area Network (RCAN-ET) Timing ..... 1095
26.3.11 Port Output Enable (POE) Timing ..... 1096
26.3.12 $I^2C$ Bus Interface 2 ($I^2C2$) Timing ..... 1097
26.3.13 UBC Trigger Timing ..... 1098
26.3.14 A/D Converter Timing ..... 1099
26.3.15 AC Characteristics Measurement Conditions ..... 1100
26.4 A/D Converter Characteristics ..... 1101
26.5 Flash Memory Characteristics ..... 1102
26.6 Usage Note ..... 1103
26.6.1 Notes on Connecting $V_{CL}$ Capacitor ..... 1103

Appendix ..... 1105
A. Pin States ..... 1105
B. Processing of Unused Pins ..... 1111
C. Pin States of Bus Related Signals ..... 1112
D. Product Code Lineup ..... 1113
E. Package Dimensions ..... 1114

Main Revisions for This Edition ..... 1117

Index ..... 1147

# Section 1 Overview

## 1.1 Features of SH7131, SH7132, SH7136, and SH7137

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

The CPU in this LSI has a RISC-type instruction set. Most instructions can be executed in one state (one system clock cycle), which greatly improves instruction execution speed. In addition, the 32-bit internal-bus architecture enhances data processing power. With this CPU, it has become possible to assemble low-cost, high-performance, and high-functioning systems, even for applications that were previously impossible with microcomputers, such as real-time control, which demands high speeds.

In addition, this LSI includes on-chip peripheral functions necessary for system configuration, such as large-capacity ROM and RAM, a data transfer controller (DTC), timers, a serial communication interface (SCI), a synchronous serial communication unit (SSU), an A/D converter, an interrupt controller (INTC), I/O ports, $I^2C$ bus interface 2 ($I^2C2$), and controller area network (RCAN-ET).

This LSI also provides an external memory access support function to enable direct connection to various memory devices or peripheral LSIs (available only with the SH7132 and SH7137).

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

The version of on-chip ROM is F-ZTAT™ (Flexible Zero Turn Around Time)* that includes flash memory. The flash memory can be programmed with a programmer that supports programming of this LSI, and can also be programmed and erased by software. This enables LSI chip to be re-programmed at a user-site while mounted on a board.

The features of this LSI are listed in table 1.1.

Note: * F-ZTAT is a trademark of Renesas Technology Corp.

**Table 1.1 Features of SH7131, SH7132, SH7136, and SH7137**

| Items | Specification |
|---|---|
| CPU | • Central processing unit with an internal 32-bit RISC (Reduced Instruction Set Computer) architecture<br>• Instruction length: 16-bit fixed length for improved code efficiency<br>• Load-store architecture (basic operations are executed between registers)<br>• Sixteen 32-bit general registers<br>• Five-stage pipeline<br>• On-chip multiplier: Multiplication operations (32 bits × 32 bits → 64 bits) executed in two to five cycles<br>• C language-oriented 62 basic instructions<br>Note: Some specifications on slot illegal instruction exception handling in this LSI differ from those of the conventional SH-2. For details, see section 5.8.4, Notes on Slot Illegal Instruction Exception Handling. |
| Operating modes | • Operating modes<br>— Single chip mode<br>— Extended ROM enabled mode (Only in SH7132/SH7137)<br>— Extended ROM disabled mode (Only in SH7132/SH7137)<br>• Operating states<br>— Program execution state<br>— Exception handling state<br>— Bus release state (Only in SH7132/SH7137)<br>• Power-down modes<br>— Sleep mode<br>— Software standby mode (Only in SH7136/SH7137)<br>— Deep software standby mode (Only in SH7136/SH7137)<br>— Module standby mode |
| User break controller (UBC)<br><br>(SH7132 and SH7137 only) | • Addresses, data values, type of access, and data size can all be set as break conditions<br>• Supports a sequential break function<br>• Two break channels |

| Items | Specification |
|---|---|
| On-chip ROM | • 128 Kbytes (Only in SH7131/SH7132) or 256 Kbytes |
| On-chip RAM | • 8 Kbytes (Only in SH7131/SH7132) or 16 Kbytes |
| Bus state controller (BSC) | • Address space: A maximum 1 Mbyte for each of two areas (CS0 and CS1) (Only in SH7132/SH7137)<br>• 8-bit external bus (Only in SH7132/SH7137)<br>• The following features settable for each area independently<br>— Number of access wait cycles<br>— Idle wait cycle insertion<br>— Supports SRAM<br>• Outputs a chip select signal according to the target area |
| Data transfer controller (DTC) | • Data transfer activated by an on-chip peripheral module interrupt can be done independently of the CPU transfer.<br>• Transfer mode selectable for each interrupt source (transfer mode is specified in memory)<br>• Multiple data transfer enabled for one activation source<br>• Various transfer modes<br>Normal mode, repeat mode, or block transfer mode can be selected.<br>• Data transfer size can be specified as byte, word, or longword<br>• The interrupt that activated the DTC can be issued to the CPU.<br>— A CPU interrupt can be requested after one data transfer completion.<br>— A CPU interrupt can be requested after all specified data transfer completion. |
| Interrupt controller (INTC) | • Five external interrupt pins (NMI and IRQ3 to IRQ0)<br>• On-chip peripheral interrupts: Priority level set for each module<br>• Vector addresses: A vector address for each interrupt source |

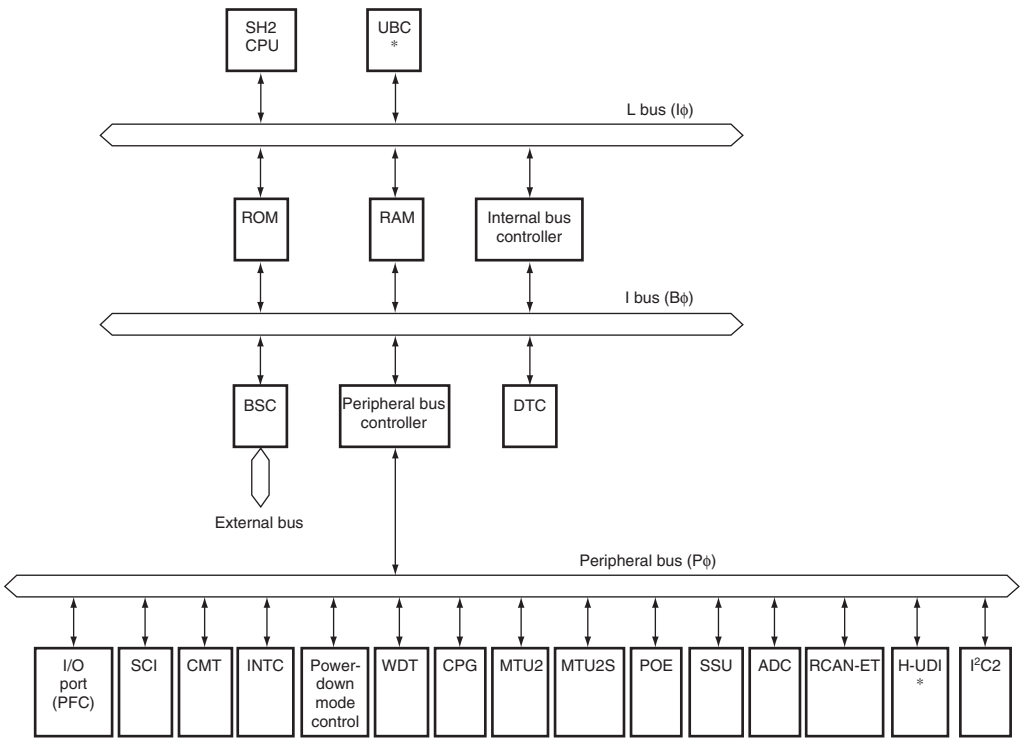
| Items | Specification |
|---|---|
| User debugging interface (H-UDI) (SH7136 and SH7137 only) | • Supports the E10A emulator |
| Clock pulse generator (CPG) | • Clock mode: Input clock can be selected from external input or crystal resonator<br>• Five types of clocks generated<br>— CPU clock: Maximum 80 MHz<br>— Bus clock: Maximum 40 MHz<br>— Peripheral clock: Maximum 40 MHz<br>— MTU2 clock: Maximum 40 MHz<br>— MTU2S clock: Maximum 80 MHz |
| Watchdog timer (WDT) | • On-chip one-channel watchdog timer<br>• Interrupt generation is supported |
| Multi-function timer pulse unit 2 (MTU2) | • Maximum 16 lines of pulse input/output and 3 lines of pulse input based on six channels of 16-bit timers<br>• 21 output compare and input capture registers<br>• A total of 21 independent comparators<br>• Selection of eight counter input clocks<br>• Input capture function<br>• Pulse output modes<br>Toggle, PWM, complementary PWM, and reset-synchronized PWM modes<br>• Synchronization of multiple counters<br>• Complementary PWM output mode<br>— Non-overlapping waveforms output for 6-phase inverter control<br>— Automatic dead time setting<br>— 0% to 100% PWM duty cycle specifiable<br>— Output suppression<br>— A/D conversion delaying function<br>— Dead time compensation function<br>— Interrupt skipping at crest or trough |

| Items | Specification |
|---|---|
| Multi-function timer pulse unit 2 (MTU2) | • Reset-synchronized PWM mode<br>Three-phase PWM waveforms in positive and negative phases can be output with a required duty cycle<br>• Phase counting mode<br>Two-phase encoder pulse counting available |
| Multi-function timer pulse unit 2S (MTU2S) | • Subset version of MTU2, having only channels 3, 4, and 5<br>• Operates at 80 MHz max. |
| Port output enable (POE) | • High-impedance control of waveform output pins in MTU2 and MTU2S |
| Compare match timer (CMT) | • 16-bit counters<br>• Compare match interrupts can be generated<br>• Two channels |
| Serial communication interface (SCI) | • Clock synchronous or asynchronous mode<br>• Three channels |
| Synchronous serial communication unit (SSU) | • Master mode or slave mode selectable<br>• Standard mode or bidirectional mode selectable<br>• Transmit/receive data length can be selected from 8, 16, and 32 bits.<br>• Full-duplex communication (transmission and reception executed simultaneously)<br>• Consecutive serial communication<br>• One channel |
| $I^2C$ bus interface (IIC2) | • Conforming to Philips $I^2C$ bus interface specifications<br>• Master mode and slave mode supported<br>• Continuous transmission/reception<br>• $I^2C$ bus format or clock synchronous serial format selectable<br>• One channel |
| Controller area network (RCAN-ET) | • CAN version: Bosch 2.0B active is supported<br>• Buffer size: 15 buffers for transmission/reception and one buffer for reception only<br>• One channel |

| Items | Specification |
|---|---|
| A/D converter (ADC) | • 12 bits × 16 channels (in SH7132/SH7137)<br>• 12 bits × 12 channels (in SH7131/SH7136)<br>• Conversion request by external triggers, MTU2, or MTU2S<br>• Two sample-and-hold function units (one unit consists of three sample-and-hold circuits) (three channels can be sampled simultaneously by an unit) |
| I/O ports | • 57 general input/output pins and 16 general input-only pins (SH7132/SH7137)<br>• 44 general input/output pins and 12 general input-only pins (SH7131/SH7136)<br>• Input or output can be selected for each bit |
| Package | • LQFP1414-100 (0.5 pitch) (SH7132/SH7137)<br>• LQFP1414-80 (0.65 pitch) (SH7131/SH7136) |
| Power supply voltage | • Vcc: 3.0 to 3.6 V or 4.0 to 5.5 V<br>• AVcc: 4.5 to 5.5 V |

## 1.2 Block Diagram

The block diagram of SH7131, SH7132, SH7136, and SH7137 is shown in figure 1.1.



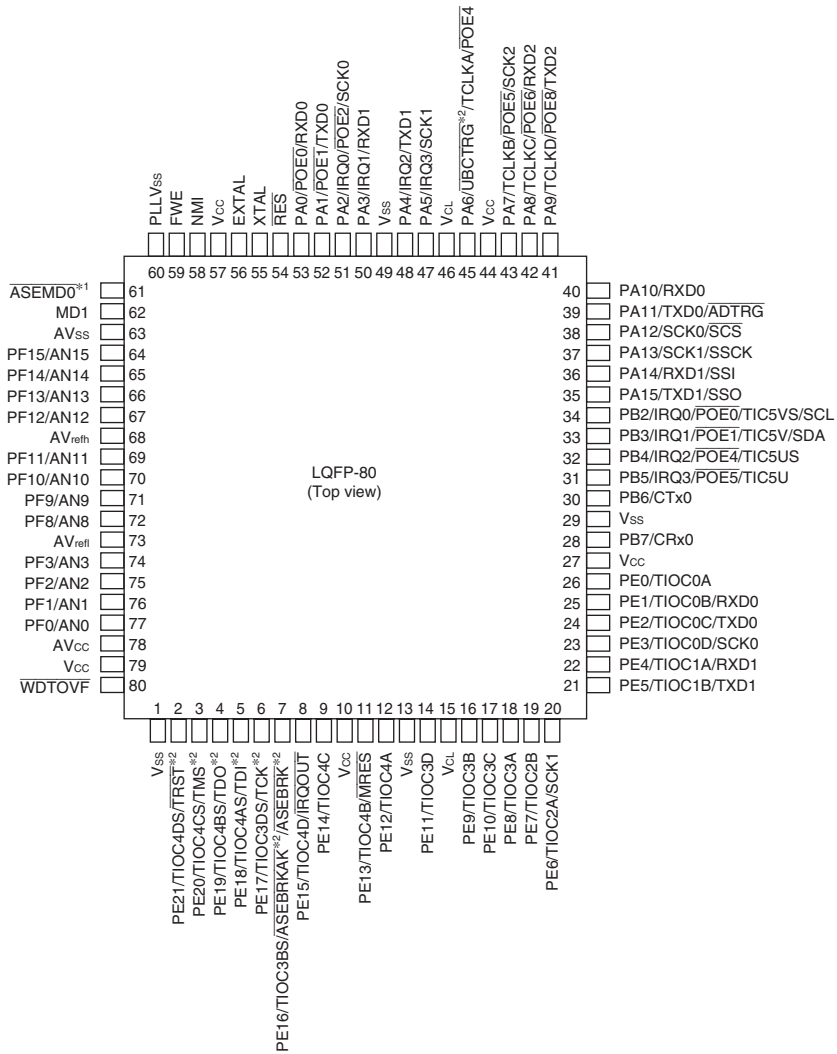

[Legend]

| | | | |
|---|---|---|---|
| ROM: | On-chip ROM | PFC: | Pin function controller |
| RAM: | On-chip RAM | MTU2: | Multi-function timer pulse unit 2 |
| UBC: | User break controller | MTU2S: | Multi-function timer pulse unit 2 (subset) |
| INTC: | Interrupt controller | POE: | Port output enable |
| CPG: | Clock pulse generator | SCI: | Serial communication interface |
| WDT: | Watchdog timer | SSU: | Synchronous serial communication unit |
| CPU: | Central processing unit | CMT: | Compare match timer |
| BSC: | Bus state controller | ADC: | A/D converter |
| DTC: | Data transfer controller | RCAN-ET: | Controller area network |
| | | $I^2C2$: | $I^2C$ Bus interface 2 |
| | | H-UDI: | User debugging interface |

Note: * SH7136 and SH7137 only.

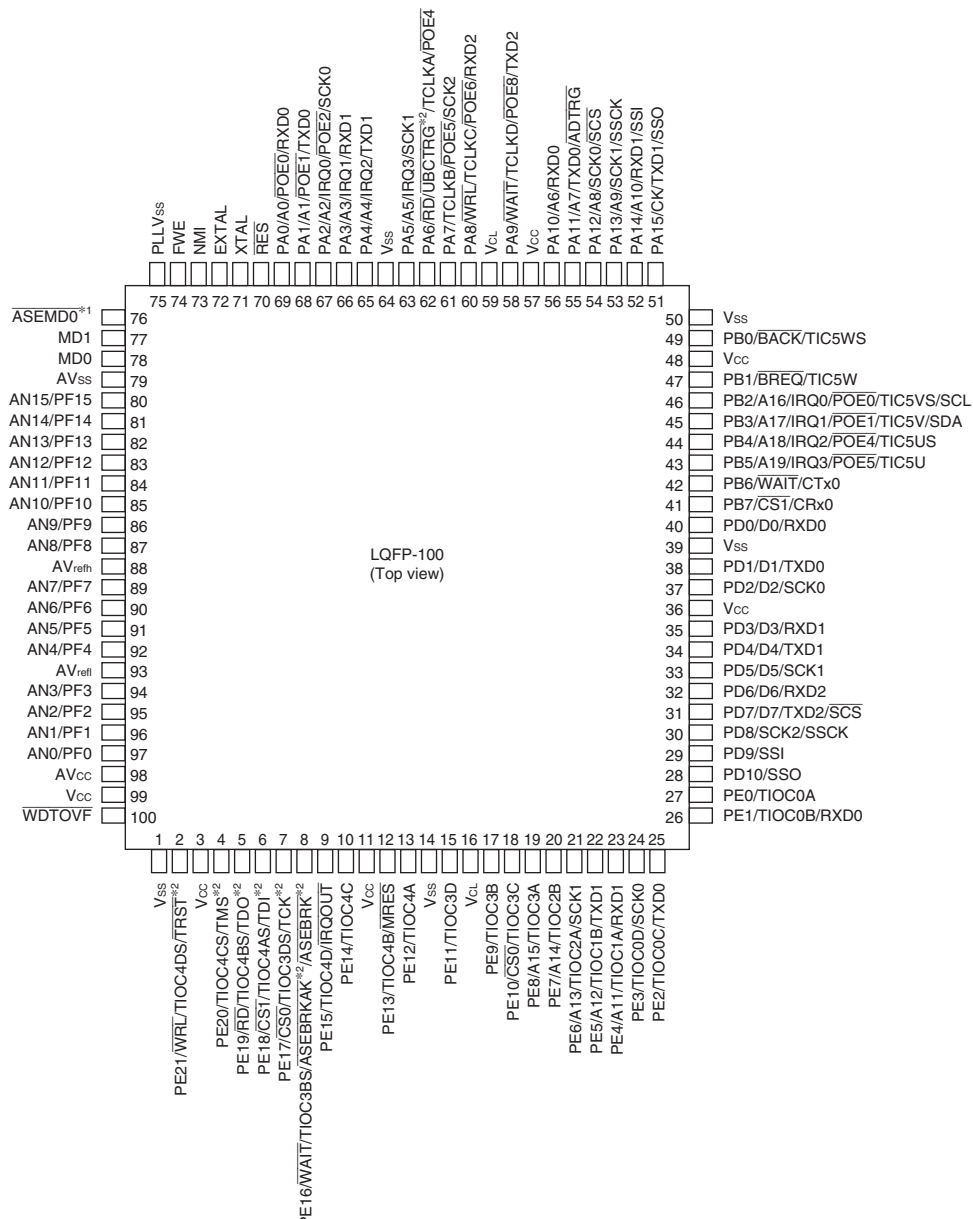
**Figure 1.1 Block Diagram**

# 1.3 Pin Assignments




Notes: 1. This pin is used by the E10A emulator. On the SH7131 it functions as a Vcc fixed pin and on the SH7136 as the ASEMD0 input pin.
2. Pin function enabled on the SH7136 only.

**Figure 1.2 SH7131 and SH7136 Pin Assignments**




Notes: 1. This pin is used by the E10A emulator. On the SH7132 it functions as a Vcc fixed pin and on the SH7137 as the ASEMD0 input pin.
2. Pin function enabled on the SH7137 only.

**Figure 1.3 SH7132 and SH7137 Pin Assignments**

## 1.4 Pin Functions

Table 1.2 summarizes the pin functions.

**Table 1.2 Pin Functions**

| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| Power supply | Vcc | I | Power supply | Power supply pins<br>Connect all Vcc pins to the system power supply. The LSI does not operate if any Vcc pins are open. |
| | Vss | I | Ground | Ground pin<br>Connect all Vss pins to the system power supply (0V). The LSI does not operate if any pins are open. |
| | $V_{CL}$ | O | Internal step-down power supply | External capacitance pins for internal step-down power supply<br>Connect these pins to Vss via a 0.47 μF capacitor (should be placed close to the pins). |
| Clock | PLLVss | I | PLL ground | Ground pin for the on-chip PLL oscillator |
| | EXTAL | I | External clock | Connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin. |
| | XTAL | O | Crystal | Connected to a crystal resonator. |
| | CK | O | System clock | Supplies the system clock to external devices. The SH7131/SH7136 does not have this pin. |

| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| Operating mode control | MD1, MD0 | I | Mode set | Sets the operating mode. Do not change values on these pins during operation. Only MD1 is available in the SH7131/SH7136. |
| | FWE | I | Flash memory write enable | Pin for flash memory<br>Flash memory can be protected against programming or erasure through this pin. |
| System control | $\overline{\text{RES}}$ | I | Power-on reset | When low, this LSI enters the power-on reset state. |
| | $\overline{\text{MRES}}$ | I | Manual reset | When low, this LSI enters the manual reset state. |
| | $\overline{\text{WDTOVF}}$ | O | Watchdog timer overflow | Output signal for the watchdog timer overflow<br>Use a resistor of 1 MΩ or more when this pin needs to be pulled down. |
| | $\overline{\text{BREQ}}$ | I | Bus-mastership request | Low when an external device requests the release of the bus mastership. This pin is available only on the SH7132/SH7137. |
| | $\overline{\text{BACK}}$ | O | Bus-mastership request acknowledge | Indicates that the bus mastership has been released to an external device. Reception of the $\overline{\text{BACK}}$ signal informs the device which has output the $\overline{\text{BREQ}}$ signal that it has acquired the bus. This pin is available only on the SH7132/SH7137. |

| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| Interrupts | NMI | I | Non-maskable interrupt | Non-maskable interrupt request pin<br>Fix to high or low level when not in use. |
| | IRQ3 to IRQ0 | I | Interrupt requests 3 to 0 | Maskable interrupt request pin<br>Selectable as level input or edge input. The rising edge, falling edge, and both edges are selectable as edges. |
| | $\overline{\text{IRQOUT}}$ | O | Interrupt request output | Shows that an interrupt cause has occurred. The interrupt cause can be recognized even in the bus release state. |
| Address bus | A19 to A0 | O | Address bus | Outputs addresses. This pin is available only on the SH7132/SH7137. |
| Data bus | D7 to D0 | I/O | Data bus | 8-bit bidirectional bus. This pin is available only on the SH7132/SH7137. |
| Bus control | $\overline{\text{CS1}}$, $\overline{\text{CS0}}$ | O | Chip select 1 and 0 | Chip-select signal for external memory or devices. This pin is available only on the SH7132/SH7137. |
| | $\overline{\text{RD}}$ | O | Read | Indicates reading of data from external devices. This pin is available only on the SH7132/SH7137. |
| | $\overline{\text{WRL}}$ | O | Write | Indicates a write access to bits 7 to 0 of the external data. This pin is available only on the SH7132/SH7137. |
| | $\overline{\text{WAIT}}$ | I | Wait | Input signal for inserting a wait cycle into the bus cycles during access to the external space. This pin is available only on the SH7132/SH7137. |

| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| 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 to 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 to 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 to TGRD_4 input capture input/output compare output/PWM output pins |
| | TIC5U, TIC5V, TIC5W | I | MTU2 input capture (channel 5) | The TGRU_5, TGRV_5, and TGRW_5 input capture input pins. (The TIC5W pin is available only on the SH7132/SH7137) |
| Multi function timer-pulse unit 2S (MTU2S) | TIOC3BS, TIOC3DS | I/O | MTU2S input capture/output compare (channel 3) | The TGRB_3S and TGRD_3S input capture input/output compare output/PWM output pins |
| | TIOC4AS, TIOC4BS, TIOC4CS, TIOC4DS | I/O | MTU2S input capture/output compare (channel 4) | The TGRA_4S to TGRD_4S input capture input/output compare output/PWM output pins |
| | TIC5US, TIC5VS, TIC5WS | I | MTU2S input capture (channel 5) | The TGRU_5S, TGRV_5S, and TGRW_5S input capture input pins (The TIC5WS pin is available only on the SH7132/SH7137) |

| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| Port output enable (POE) | $\overline{\text{POE8}}$, $\overline{\text{POE6}}$ to $\overline{\text{POE4}}$, $\overline{\text{POE2}}$ to $\overline{\text{POE0}}$ | I | Port output enable | Request signal input to place the MTU2 and MTU2S waveform output pins in high impedance state. |
| Serial communication interface (SCI) | TXD2 to TXD0 | O | Transmit data | Transmit data output pins |
| | RXD2 to RXD0 | I | Receive data | Receive data input pins |
| | SCK2 to SCK0 | I/O | Serial clock | Clock input/output pins |
| Synchronous serial communication unit (SSU) | SSO | I/O | Data | Data input/output pin |
| | SSI | I/O | Data | Data input/output pin |
| | SSCK | I/O | Clock | Clock input/output pin |
| | $\overline{\text{SCS}}$ | I/O | Chip select | Chip select input/output pin |
| Controller area network (RCAN-ET) | CTx0 | O | Transmit data | Transmit data pin for CAN bus |
| | CRx0 | I | Receive data | Receive data pin for CAN bus |
| $I^2C$ bus interface 2 ($I^2C2$) | SCL | I/O | $I^2C$ clock input/output | $I^2C$ bus clock input/output pin |
| | SDA | I/O | $I^2C$ data input/output | $I^2C$ bus data input/output pin |
| A/D converter (ADC) | AN15 to AN0 | I | Analog input pins | Analog input pins (AN15 to AN8, AN3 to AN0 for the SH7131/SH7136) |
| | $\overline{\text{ADTRG}}$ | I | A/D conversion trigger input | External trigger input pin for starting A/D conversion |
| | AVcc | I | Analog power supply | Power supply pin for the A/D converter<br>Connect it to the system power supply (Vcc) when the A/D converter is not used.<br>Connect all AVcc pins to the system power supply (Vcc) The A/D converter does not work if any pin is open. |

| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| A/D converter (ADC) | AVss | I | Analog ground | Ground pin for the A/D converter<br>Connect it to the system ground (0 V).<br>Connect all AVss pins to the system ground (0 V) correctly. The A/D converter does not work if any pin is open. |
| | AVrefh | I | Analog reference power supply (high) | Analog reference power supply (high) |
| | AVrefl | I | Analog reference power supply (low) | Analog reference power supply (low) |
| I/O ports | PA15 to PA0 | I/O | General port | 16 bits of general input/output port pins |
| | PB7 to PB0 | I/O | General port | 8 bits of general input/output port pins (PB7 to PB2 in the SH7131/SH7136) |
| | PD10 to PD0 | I/O | General port | 11 bits of general input/output port pins (The SH7131/SH7136 does not have this pin) |
| | PE21 to PE0 | I/O | General port | 22 bits of general input/output port pins |
| | PF15 to PF0 | I | General port | 16 bits of general input port pins (PF15 to PF8, PF3 to PF0 in the SH7131/SH7136) |
| User break controller (UBC) | $\overline{\text{UBCTRG}}$ | O | User break trigger output | Trigger output pin for UBC condition match<br>Available only in the SH7136/ SH7137. |
| User debugging interface (H-UDI) (SH7136 and SH7137 only) | TCK | I | Test clock | Test-clock input pin. |
| | TMS | I | Test mode select | Inputs the test-mode select signal. |
| | TDI | I | Test data input | Serial input pin for instructions and data. |
| | TDO | O | Test data output | Serial output pin for instructions and data. |
| | $\overline{\text{TRST}}$ | I | Test reset | Initialization-signal input pin. |

| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| E10A interface (SH7136 and SH7137 only) | $\overline{\text{ASEMDO}}$ | I | ASE mode | Sets the ASE mode. When this pin is driven low, the LSI enters ASE mode, and when driven high, the LSI operates in normal mode. Emulator dedicated functions can be used in the ASE mode. When nothing is input to this pin, it is pulled up internally. |
| | $\overline{\text{ASEBRK}}$ | I | Break request | E10A emulator break input |
| | $\overline{\text{ASEBRKAK}}$ | O | Break mode acknowledge | Indicates the E10A emulator has entered the break mode. |

Note: The $\overline{\text{WDTOVF}}$ pin should not be pulled down. If it must be pulled down, use a resistor of 1 MΩ or more.

# Section 2 CPU

## 2.1 Features

- General registers: 32-bit register × 16
- Basic instructions: 62
- Addressing modes: 11

  Register direct (Rn)

  Register indirect (@Rn)

  Post-increment register indirect (@Rn+)

  Pre-decrement register indirect (@-Rn)

  Register indirect with displacement (@disp:4, Rn)

  Index register indirect (@R0, Rn)

  GBR indirect with displacement (@disp:8, GBR)

  Index GBR indirect (@R0, GBR)

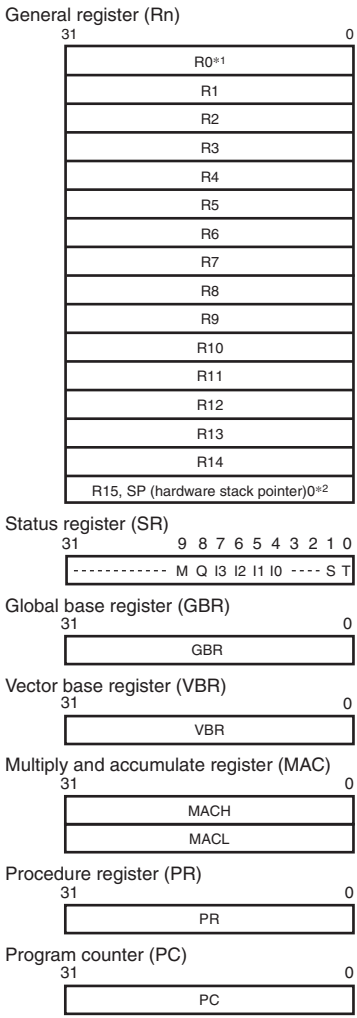
  PC relative with displacement (@disp:8, PC)

  PC relative (disp:8/disp:12/Rn)

  Immediate (#imm:8)

## 2.2 Register Configuration

There are three types of registers: general registers (32-bit × 16), control registers (32-bit × 3), and system registers (32-bit × 4).




Notes: 1. R0 can be used as an index register in index register indirect or index GBR indirect addressing mode. For some instructions, only R0 is used as the source or destination register.
2. R15 is used as a hardware stack pointer during exception handling.

**Figure 2.1 CPU Internal Register Configuration**

### 2.2.1 General Registers (Rn)

There are sixteen 32-bit general registers (Rn), designated R0 to R15. The general registers are used for data processing and address calculation. R0 is also used as an index register. With a number of instructions, R0 is the only register that can be used. R15 is used as a hardware stack pointer (SP). In exception handling, R15 is used for accessing the stack to save or restore the status register (SR) and program counter (PC) values.

### 2.2.2 Control Registers

There are three 32-bit control registers, designated status register (SR), global base register (GBR), and vector base register (VBR). SR indicates a processing state. GBR is used as a base address in GBR indirect addressing mode for data transfer of on-chip peripheral module registers. VBR is used as a base address of the exception handling (including interrupts) vector table.

- 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | 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 | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W |

| Bit | Bit name | Default | Read/ Write | Description |
|---|---|---|---|---|
| 31 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 9 | M | Undefined | R/W | Used by the DIV0U, DIV0S, and DIV1 instructions. |
| 8 | Q | Undefined | R/W | Used by the DIV0U, DIV0S, and DIV1 instructions. |
| 7 to 4 | I[3:0] | 1111 | R/W | Interrupt Mask |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit name | Default | Read/ Write | Description |
|---|---|---|---|---|
| 1 | S | Undefined | R/W | S Bit<br>Used by the multiply and accumulate instruction. |
| 0 | T | Undefined | R/W | T Bit<br>Indicates true (1) or false (0) in the following instructions: MOVT, CMP/cond, TAS, TST, BT (BT/S), BF (BF/S), SETT, CLRT<br>Indicates carry, borrow, overflow, or underflow in the following instructions: ADDV, ADDC, SUBV, SUBC, NEGC, DIV0U, DIV0S, DIV1, SHAR, SHAL, SHLR, SHLL, ROTR, ROTL, ROTCR, ROTCL |

- Global-base register (GBR)

  This register indicates a base address in GBR indirect addressing mode. The GBR indirect addressing mode is used for data transfer of the on-chip peripheral module registers and logic operations.
- Vector-base register (VBR)

  This register indicates the base address of the exception handling vector table.

### 2.2.3 System Registers

There are four 32-bit system registers, designated two multiply and accumulate registers (MACH and MACL), a procedure register (PR), and program counter (PC).

- Multiply and accumulate registers (MACH and MACL)
  This register stores the results of multiplication and multiply-and-accumulate operation.
- Procedure register (PR)
  This register stores the return-destination address from subroutine procedures.
- Program counter (PC)
  The PC indicates the point which is four bytes (two instructions) after the current execution instruction.

### 2.2.4 Initial Values of Registers

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

**Table 2.1 Initial Values of Registers**

| Type of register | Register | Default |
|---|---|---|
| General register | R0 to R14 | Undefined |
| | R15 (SP) | SP value set in the exception handling vector table |
| Control register | SR | I3 to I0: 1111 (H'F)<br>Reserved bits: 0<br>Other bits: Undefined |
| | GBR | Undefined |
| | VBR | H'00000000 |
| System register | MACH, MACL, PR | Undefined |
| | PC | PC value set in the exception handling vector table |

## 2.3 Data Formats

### 2.3.1 Register Data Format

The size of register operands is always longwords (32 bits). When loading byte (8 bits) or word (16 bits) data in memory into a register, the data is sign-extended to longword and stored in the register.

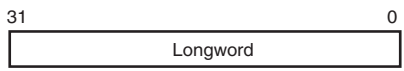



**Figure 2.2 Register Data Format**

### 2.3.2 Memory Data Formats

Memory data formats are classified into bytes, words, and longwords. Byte data can be accessed from any address. Locate, however, word data at an address 2n, longword data at 4n. Otherwise, an address error will occur if an attempt is made to access word data starting from an address other than 2n or longword data starting from an address other than 4n. In such cases, the data accessed cannot be guaranteed. The hardware stack area, pointed by the hardware stack pointer (SP, R15), uses only longword data starting from address 4n because this area holds the program counter and status register.

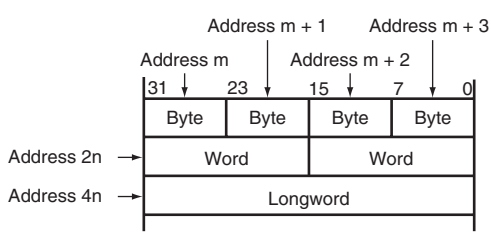



**Figure 2.3 Memory Data Format**

### 2.3.3 Immediate Data Formats

Immediate data of eight bits is placed in the instruction code.

For the MOV, ADD, and CMP/EQ instructions, the immediate data is sign-extended to longword and then calculated. For the TST, AND, OR, and XOR instructions, the immediate data is zero-extended to longword and then calculated. Thus, if the immediate data is used for the AND instruction, the upper 24 bits in the destination register are always cleared.

The immediate data of word or longword is not placed in the instruction code. It is placed in a table in memory. The table in memory is accessed by the MOV immediate data instruction in PC relative addressing mode with displacement.

## 2.4 Features of Instructions

### 2.4.1 RISC Type

The instructions are RISC-type instructions with the following features:

**Fixed 16-Bit Length:** All instructions have a fixed length of 16 bits. This improves program code efficiency.

**One Instruction per Cycle:** Since pipelining is used, basic instructions can be executed in one cycle.

**Data Size:** The basic data size for operations is longword. Byte, word, or longword can be selected as the memory access size. Byte or word data in memory is sign-extended to longword and then calculated. Immediate data is sign-extended to longword for arithmetic operations or zero-extended to longword size for logical operations.

**Table 2.2 Word Data Sign Extension**

| CPU in this LSI | Description | Example of Other CPUs |
|---|---|---|
| MOV.W @(disp,PC),R1<br>ADD R1,R0<br>.....<br>.DATA.W H'1234 | Sign-extended to 32 bits, R1 becomes H'00001234, and is then operated on by the ADD instruction. | ADD.W #H'1234,R0 |

Note: Immediate data is accessed by @(disp,PC).

**Load/Store Architecture:** Basic operations are executed between registers. In operations involving memory, data is first loaded into a register (load/store architecture). However, bit manipulation instructions such as AND are executed directly in memory.

**Delayed Branching:** Unconditional branch instructions means the delayed branch instructions. With a delayed branch instruction, the branch is made after execution of the instruction immediately following the delayed branch instruction. This minimizes disruption of the pipeline when a branch is made. The conditional branch instructions have two types of instructions: conditional branch instructions and delayed branch instructions.

**Table 2.3 Delayed Branch Instructions**

| CPU in this LSI | | Description | Example of Other CPUs | |
|---|---|---|---|---|
| BRA | TRGET | ADD is executed before branch to TRGET. | ADD.W | R1,R0 |
| ADD | R1,R0 | | BRA | TRGET |

**Multiply/Multiply-and-Accumulate Operations:** A 16 × 16 → 32 multiply operation is executed in one to two cycles, and a 16 × 16 + 64 → 64 multiply-and-accumulate operation in two to three cycles. A 32 × 32 → 64 multiply operation and a 32 × 32 + 64 → 64 multiply-and-accumulate operation are each executed in two to four cycles.

**T Bit:** The result of a comparison is indicated by the T bit in SR, and a conditional branch is performed according to whether the result is True or False. Processing speed has been improved by keeping the number of instructions that modify the T bit to a minimum.

**Table 2.4 T Bit**

| CPU in this LSI | | Description | Example of Other CPUs | |
|---|---|---|---|---|
| CMP/GE | R1,R0 | When R0 ≥ R1, the T bit is set. | CMP.W | R1,R0 |
| BT | TRGET0 | When R0 ≥ R1, a branch is made to TRGET0. | BGE | TRGET0 |
| BF | TRGET1 | When R0 < R1, a branch is made to TRGET1. | BLT | TRGET1 |
| ADD | #–1,R0 | The T bit is not changed by ADD. | SUB.W | #1,R0 |
| CMP/EQ | #0,R0 | When R0 = 0, the T bit is set. | BEQ | TRGET |
| BT | TRGET | A branch is made when R0 = 0. | | |

**Immediate Data:** 8-bit immediate data is placed in the instruction code. Word and longword immediate data is not placed in the instruction code. It is placed in a table in memory. The table in memory is accessed with the MOV immediate data instruction using PC relative addressing mode with displacement.

**Table 2.5 Access to Immediate Data**

| Type | This LSI's CPU | Example of Other CPU |
|---|---|---|
| 8-bit immediate | MOV #H'12,R0 | MOV.B #H'12,R0 |
| 16-bit immediate | MOV.W @(disp,PC),R0<br>.....<br>.DATA.W H'1234 | MOV.W #H'1234,R0 |
| 32-bit immediate | MOV.L @(disp,PC),R0<br>.....<br>.DATA.L H'12345678 | MOV.L #H'12345678,R0 |

Note: Immediate data is accessed by @(disp,PC).

**Absolute Addresses:** When data is accessed by absolute address, place the absolute address value in a table in memory beforehand. The absolute address value is transferred to a register using the method whereby immediate data is loaded when an instruction is executed, and the data is accessed using the register indirect addressing mode.

**Table 2.6 Access to Absolute Address**

| Type | CPU in this LSI | Example of Other CPUs |
|---|---|---|
| Absolute address | MOV.L @(disp,PC),R1<br>MOV.B @R1,R0<br>.....<br>.DATA.L H'12345678 | MOV.B @H'12345678,R0 |

Note: Immediate data is referenced by @(disp,PC).

**16-Bit/32-Bit Displacement:** When data is accessed using the 16- or 32-bit displacement addressing mode, the displacement value is placed in a table in memory beforehand. Using the method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is accessed using index register indirect addressing mode.

**Table 2.7 Access with Displacement**

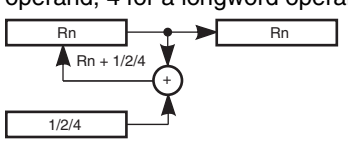
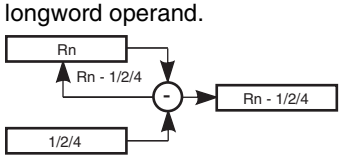
| Type | CPU in this LSI | Example of Other CPUs |
|---|---|---|
| 16-bit displacement | MOV.W @(disp,PC),R0<br>MOV.W @(R0,R1),R2<br>.....<br>.DATA.W H'1234 | MOV.W @(H'1234,R1),R2 |

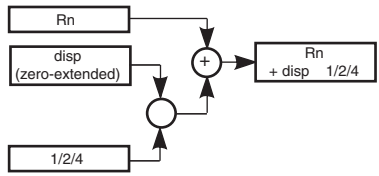
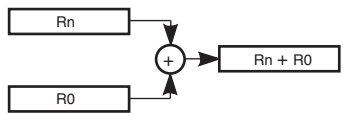
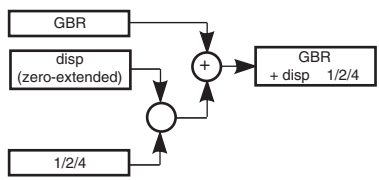
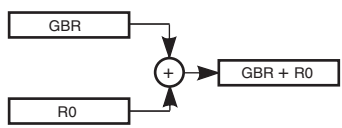
Note: Immediate data is referenced by @(disp,PC).

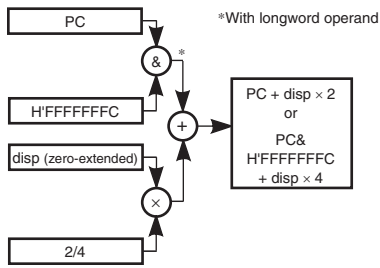
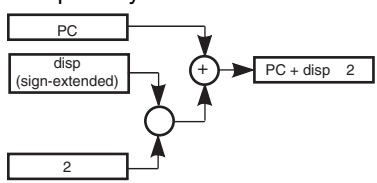
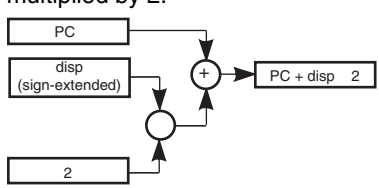
### 2.4.2 Addressing Modes

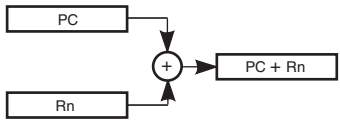
Table 2.8 lists addressing modes and effective address calculation methods.

**Table 2.8 Addressing Modes and Effective Addresses**

| Addressing Mode | Instruction Format | Effective Address Calculation Method | Calculation Formula |
|---|---|---|---|
| Register direct | Rn | Effective address is register Rn.<br>(Operand is register Rn contents.) | — |
| Register indirect | @Rn | Effective address is register Rn contents.<br> | Rn |
| Register indirect with post-increment | @Rn+ | Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand.<br>  | Rn<br>After instruction execution<br>Byte: Rn + 1 → Rn<br>Word: Rn + 2 → Rn<br>Longword: Rn + 4 → Rn |
| Register indirect with pre-decrement | @–Rn | Effective address is register Rn contents, decremented by a constant beforehand: 1 for a byte operand, 2 for a word operand, 4 for a longword operand.<br>  | Byte: Rn – 1 → Rn<br>Word: Rn – 2 → Rn<br>Longword: Rn – 4 → Rn<br>(Instruction executed with Rn after calculation) |

| Addressing Mode | Instruction Format | Effective Address Calculation Method | Calculation Formula |
|---|---|---|---|
| Register indirect with displacement | @(disp:4, Rn) | Effective address is register Rn contents with 4-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size.   | Byte: Rn + disp<br>Word: Rn + disp × 2<br>Longword: Rn + disp × 4 |
| Index register indirect | @(R0, Rn) | Effective address is sum of register Rn and R0 contents.   | Rn + R0 |
| GBR indirect with displacement | @(disp:8, GBR) | Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size.   | Byte: GBR + disp<br>Word: GBR + disp × 2<br>Longword: GBR + disp × 4 |
| Index GBR indirect | @(R0, GBR) | Effective address is sum of register GBR and R0 contents.   | GBR + R0 |

| Addressing Mode | Instruction Format | Effective Address Calculation Method | Calculation Formula |
|---|---|---|---|
| PC relative with displacement | @(disp:8, PC) | Effective address is PC with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word) or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked.   | Word: PC + disp × 2<br>Longword: PC&H'FFFFFFFC + disp × 4 |
| PC relative | disp:8 | Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2.   | PC + disp × 2 |
| | disp:12 | Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2.   | PC + disp × 2 |

| Addressing Mode | Instruction Format | Effective Address Calculation Method | Calculation Formula |
|---|---|---|---|
| PC relative | Rn | Effective address is sum of PC and Rn.   | PC + Rn |
| Immediate | #imm:8 | 8-bit immediate data imm of TST, AND, OR, or XOR instruction is zero-extended. | — |
| | #imm:8 | 8-bit immediate data imm of MOV, ADD, or CMP/EQ instruction is sign-extended. | — |
| | #imm:8 | 8-bit immediate data imm of TRAPA instruction is zero-extended and multiplied by 4. | — |

### 2.4.3 Instruction Formats

This section describes the instruction formats, and the meaning of the source and destination operands. The meaning of the operands depends on the instruction code. The following symbols are used in the table.

`xxxx:` Instruction code

`mmmm:` Source register

`nnnn:` Destination register

`iiii:` Immediate data

`dddd:` Displacement

**Table 2.9 Instruction Formats**

| Instruction Format | Source Operand | Destination Operand | Sample Instruction |
|---|---|---|---|
| 0 type<br>15 0<br>xxxx xxxx xxxx xxxx | — | — | NOP |
| n type<br>15 0<br>xxxx nnnn xxxx xxxx | — | nnnn: register direct | MOVT Rn |
| | Control register or system register | nnnn: register direct | STS MACH,Rn |
| | Control register or system register | nnnn: pre-decrement register indirect | STC.L SR,@-Rn |
| m type<br>15 0<br>xxxx mmmm xxxx xxxx | mmmm: register direct | Control register or system register | LDC Rm,SR |
| | mmmm: post-increment register indirect | Control register or system register | LDC.L @Rm+,SR |
| | mmmm: register indirect | — | JMP @Rm |
| | PC relative using Rm | — | BRAF Rm |

| Instruction Format | Source Operand | Destination Operand | Sample Instruction |
|---|---|---|---|
| nm type<br>15 0<br>xxxx nnnn mmmm xxxx | mmmm: register direct | nnnn: register direct | ADD Rm,Rn |
| | mmmm: register direct | nnnn: register indirect | MOV.L Rm,@Rn |
| | mmmm: post-increment register indirect (multiply-and-accumulate operation)<br>nnnn: * post-increment register indirect (multiply-and-accumulate operation) | MACH, MACL | MAC.W @Rm+,@Rn+ |
| | mmmm: post-increment register indirect | nnnn: register direct | MOV.L @Rm+,Rn |
| | mmmm: register direct | nnnn: pre-decrement register indirect | MOV.L Rm,@-Rn |
| | mmmm: register direct | nnnn: index register indirect | MOV.L Rm,@(R0,Rn) |
| md type<br>15 0<br>xxxx xxxx mmmm dddd | mmmmdddd: register indirect with displacement | R0 (register direct) | MOV.B @(disp,Rm),R0 |
| nd4 type<br>15 0<br>xxxx xxxx nnnn dddd | R0 (register direct) | nnnndddd: register indirect with displacement | MOV.B R0,@(disp,Rn) |
| nmd type<br>15 0<br>xxxx nnnn mmmm dddd | 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 |

| Instruction Format | Source Operand | Destination Operand | Sample Instruction |
|---|---|---|---|
| d type<br>15 0<br>xxxx xxxx dddd dddd | 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: PC relative | BF label |
| d12 type<br>15 0<br>xxxx dddd dddd dddd | — | dddddddddddd: PC relative | BRA label<br>(label=disp+PC) |
| nd8 type<br>15 0<br>xxxx nnnn dddd dddd | dddddddd: PC relative with displacement | nnnn: register direct | MOV.L @(disp,PC),Rn |
| i type<br>15 0<br>xxxx xxxx iiii iiii | iiiiiiii: immediate | Index GBR indirect | AND.B #imm,@(R0,GBR) |
| | iiiiiiii: immediate | R0 (register direct) | AND #imm,R0 |
| | iiiiiiii: immediate | — | TRAPA #imm |
| ni type<br>15 0<br>xxxx nnnn iiii iiii | iiiiiiii: immediate | nnnn: register direct | ADD #imm,Rn |

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

## 2.5 Instruction Set

### 2.5.1 Instruction Set by Type

Table 2.10 lists the instructions classified by type.

**Table 2.10 Instruction Types**

| Type | Kinds of Instruction | Op Code | Function | Number of Instructions |
|---|---|---|---|---|
| Data transfer instructions | 5 | MOV | Data transfer<br>Immediate data transfer<br>Peripheral module data transfer<br>Structure data transfer | 39 |
| | | MOVA | Effective address transfer | |
| | | MOVT | T bit transfer | |
| | | SWAP | Upper/lower swap | |
| | | XTRCT | Extraction of middle of linked registers | |
| Arithmetic operation instructions | 21 | ADD | Binary addition | 33 |
| | | ADDC | Binary addition with carry | |
| | | ADDV | Binary addition with overflow | |
| | | CMP/cond | Comparison | |
| | | DIV1 | Division | |
| | | DIV0S | Signed division initialization | |
| | | DIV0U | Unsigned division initialization | |
| | | 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 | |
| | | MUL | Double-precision multiplication | |

| Type | Kinds of Instruction | Op Code | Function | Number of Instructions |
|---|---|---|---|---|
| Arithmetic operation instructions | 21 | MULS | Signed multiplication | 33 |
| | | MULU | Unsigned multiplication | |
| | | NEG | Sign inversion | |
| | | NEGC | Sign inversion with borrow | |
| | | SUB | Binary subtraction | |
| | | SUBC | Binary subtraction with carry | |
| | | SUBV | Binary subtraction with underflow | |
| Logic operation instructions | 6 | AND | Logical AND | 14 |
| | | NOT | Bit inversion | |
| | | OR | Logical OR | |
| | | TAS | Memory test and bit setting | |
| | | TST | T bit setting for logical AND | |
| | | XOR | Exclusive logical OR | |
| Shift instructions | 10 | ROTL | 1-bit left shift | 14 |
| | | ROTR | 1-bit right shift | |
| | | ROTCL | 1-bit left shift with T bit | |
| | | ROTCR | 1-bit right shift with T bit | |
| | | SHAL | Arithmetic 1-bit left shift | |
| | | SHAR | Arithmetic 1-bit right shift | |
| | | SHLL | Logical 1-bit left shift | |
| | | SHLLn | Logical n-bit left shift | |
| | | SHLR | Logical 1-bit right shift | |
| | | SHLRn | Logical n-bit right shift | |

| Type | Kinds of Instruction | Op Code | Function | Number of Instructions |
|---|---|---|---|---|
| Branch instructions | 9 | BF | Conditional branch, delayed conditional branch (T = 0) | 11 |
| | | BT | Conditional branch, delayed conditional branch (T = 1) | |
| | | BRA | Unconditional branch | |
| | | BRAF | Unconditional branch | |
| | | BSR | Branch to subroutine procedure | |
| | | BSRF | Branch to subroutine procedure | |
| | | JMP | Unconditional branch | |
| | | JSR | Branch to subroutine procedure | |
| | | RTS | Return from subroutine procedure | |
| System control instructions | 11 | CLRT | T bit clear | 31 |
| | | CLRMAC | MAC register clear | |
| | | LDC | Load into control register | |
| | | LDS | Load into system register | |
| | | NOP | No operation | |
| | | RTE | Return from exception handling | |
| | | SETT | T bit setting | |
| | | SLEEP | Transition to power-down mode | |
| | | STC | Store from control register | |
| | | STS | Store from system register | |
| | | TRAPA | Trap exception handling | |
| Total: | 62 | | | 142 |

The instruction code, operation, and execution cycles of the instructions are listed in the following tables, classified by type.

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

Notes: 1. The table shows the minimum number of execution states. In practice, the number of instruction execution states will be increased in cases such as the following:
- When there is contention between an instruction fetch and a data access
- When the destination register of a load instruction (memory → register) is also used by the following instruction

2. Scaled (×1, ×2, or ×4) according to the instruction operand size, etc.
For details, see SH-1/SH-2/SH-DSP Software Manual.

### 2.5.2 Data Transfer Instructions

**Table 2.11 Data Transfer Instructions**

| Instruction | | Operation | Code | Execution Cycles | T Bit |
|---|---|---|---|---|---|
| MOV | #imm,Rn | imm → Sign extension → Rn | 1110nnnniiiiiiii | 1 | — |
| MOV.W | @(disp,PC),Rn | (disp × 2 + PC) → Sign extension → Rn | 1001nnnnddddddd | 1 | — |
| MOV.L | @(disp,PC),Rn | (disp × 4 + PC) → Rn | 1101nnnnddddddd | 1 | — |
| MOV | Rm,Rn | Rm → Rn | 0110nnnnmmmm0011 | 1 | — |
| MOV.B | Rm,@Rn | Rm → (Rn) | 0010nnnnmmmm0000 | 1 | — |
| MOV.W | Rm,@Rn | Rm → (Rn) | 0010nnnnmmmm0001 | 1 | — |
| MOV.L | Rm,@Rn | Rm → (Rn) | 0010nnnnmmmm0010 | 1 | — |
| MOV.B | @Rm,Rn | (Rm) → Sign extension → Rn | 0110nnnnmmmm0000 | 1 | — |
| MOV.W | @Rm,Rn | (Rm) → Sign extension → Rn | 0110nnnnmmmm0001 | 1 | — |
| MOV.L | @Rm,Rn | (Rm) → Rn | 0110nnnnmmmm0010 | 1 | — |
| MOV.B | Rm,@–Rn | Rn–1 → Rn, Rm → (Rn) | 0010nnnnmmmm0100 | 1 | — |
| MOV.W | Rm,@–Rn | Rn–2 → Rn, Rm → (Rn) | 0010nnnnmmmm0101 | 1 | — |
| MOV.L | Rm,@–Rn | Rn–4 → Rn, Rm → (Rn) | 0010nnnnmmmm0110 | 1 | — |
| MOV.B | @Rm+,Rn | (Rm) → Sign extension → Rn, Rm + 1 → Rm | 0110nnnnmmmm0100 | 1 | — |
| MOV.W | @Rm+,Rn | (Rm) → Sign extension → Rn, Rm + 2 → Rm | 0110nnnnmmmm0101 | 1 | — |
| MOV.L | @Rm+,Rn | (Rm) → Rn,Rm + 4 → Rm | 0110nnnnmmmm0110 | 1 | — |
| MOV.B | R0,@(disp,Rn) | R0 → (disp + Rn) | 10000000nnnndddd | 1 | — |
| MOV.W | R0,@(disp,Rn) | R0 → (disp × 2 + Rn) | 10000001nnnndddd | 1 | — |
| MOV.L | Rm,@(disp,Rn) | Rm → (disp × 4 + Rn) | 0001nnnnmmmmdddd | 1 | — |
| MOV.B | @(disp,Rm),R0 | (disp + Rm) → Sign extension → R0 | 10000100mmmmdddd | 1 | — |
| MOV.W | @(disp,Rm),R0 | (disp × 2 + Rm) → Sign extension → R0 | 10000101mmmmdddd | 1 | — |
| MOV.L | @(disp,Rm),Rn | (disp × 4 + Rm) → Rn | 0101nnnnmmmmdddd | 1 | — |

| Instruction | Operation | Code | Execution Cycles | T Bit |
|---|---|---|---|---|
| MOV.B Rm,@(R0,Rn) | Rm → (R0 + Rn) | 0000nnnnmmmm0100 | 1 | — |
| MOV.W Rm,@(R0,Rn) | Rm → (R0 + Rn) | 0000nnnnmmmm0101 | 1 | — |
| MOV.L Rm,@(R0,Rn) | Rm → (R0 + Rn) | 0000nnnnmmmm0110 | 1 | — |
| MOV.B @(R0,Rm),Rn | (R0 + Rm) → Sign extension → Rn | 0000nnnnmmmm1100 | 1 | — |
| MOV.W @(R0,Rm),Rn | (R0 + Rm) → Sign extension → Rn | 0000nnnnmmmm1101 | 1 | — |
| MOV.L @(R0,Rm),Rn | (R0 + Rm) → Rn | 0000nnnnmmmm1110 | 1 | — |
| MOV.B R0,@(disp,GBR) | R0 → (disp + GBR) | 11000000dddddddd | 1 | — |
| MOV.W R0,@(disp,GBR) | R0 → (disp × 2 + GBR) | 11000001dddddddd | 1 | — |
| MOV.L R0,@(disp,GBR) | R0 → (disp × 4 + GBR) | 11000010dddddddd | 1 | — |
| MOV.B @(disp,GBR),R0 | (disp + GBR) → Sign extension → R0 | 11000100dddddddd | 1 | — |
| MOV.W @(disp,GBR),R0 | (disp × 2 + GBR) → Sign extension → R0 | 11000101dddddddd | 1 | — |
| MOV.L @(disp,GBR),R0 | (disp × 4 + GBR) → R0 | 11000110dddddddd | 1 | — |
| MOVA @(disp,PC),R0 | disp × 4 + PC → R0 | 11000111dddddddd | 1 | — |
| MOVT Rn | T → Rn | 0000nnnn00101001 | 1 | — |
| SWAP.B Rm,Rn | Rm → Swap lowest two bytes → Rn | 0110nnnnmmmm1000 | 1 | — |
| SWAP.W Rm,Rn | Rm → Swap two consecutive words → Rn | 0110nnnnmmmm1001 | 1 | — |
| XTRCT Rm,Rn | Rm: Middle 32 bits of Rn → Rn | 0010nnnnmmmm1101 | 1 | — |

### 2.5.3 Arithmetic Operation Instructions

**Table 2.12 Arithmetic Operation Instructions**

| Instruction | | Operation | Code | Execution Cycles | T Bit |
|---|---|---|---|---|---|
| ADD | Rm,Rn | Rn + Rm → Rn | 0011nnnnmmmm1100 | 1 | — |
| ADD | #imm,Rn | Rn + imm → Rn | 0111nnnniiiiiiii | 1 | — |
| ADDC | Rm,Rn | Rn + Rm + T → Rn, Carry → T | 0011nnnnmmmm1110 | 1 | Carry |
| ADDV | Rm,Rn | Rn + Rm → Rn, Overflow → T | 0011nnnnmmmm1111 | 1 | Overflow |
| CMP/EQ | #imm,R0 | If R0 = imm, 1 → T | 10001000iiiiiiii | 1 | Comparison result |
| CMP/EQ | Rm,Rn | If Rn = Rm, 1 → T | 0011nnnnmmmm0000 | 1 | Comparison result |
| CMP/HS | Rm,Rn | If Rn ≥ Rm with unsigned data, 1 → T | 0011nnnnmmmm0010 | 1 | Comparison result |
| CMP/GE | Rm,Rn | If Rn ≥ Rm with signed data, 1 → T | 0011nnnnmmmm0011 | 1 | Comparison result |
| CMP/HI | Rm,Rn | If Rn > Rm with unsigned data, 1 → T | 0011nnnnmmmm0110 | 1 | Comparison result |
| CMP/GT | Rm,Rn | If Rn > Rm with signed data, 1 → T | 0011nnnnmmmm0111 | 1 | Comparison result |
| CMP/PZ | Rn | If Rn ≥ 0, 1 → T | 0100nnnn00010001 | 1 | Comparison result |
| CMP/PL | Rn | If Rn > 0, 1 → T | 0100nnnn00010101 | 1 | Comparison result |
| CMP/STR | Rm,Rn | If Rn and Rm have an equivalent byte, 1 → T | 0010nnnnmmmm1100 | 1 | Comparison result |
| DIV1 | Rm,Rn | Single-step division (Rn/Rm) | 0011nnnnmmmm0100 | 1 | Calculation result |
| DIV0S | Rm,Rn | MSB of Rn → Q, MSB of Rm → M, M^ Q → T | 0010nnnnmmmm0111 | 1 | Calculation result |
| DIV0U | | 0 → M/Q/T | 0000000000011001 | 1 | 0 |
| DMULS.L | Rm,Rn | Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits | 0011nnnnmmmm1101 | 2 to 5* | — |

| Instruction | | Operation | Code | Execution Cycles | T Bit |
|---|---|---|---|---|---|
| DMULU.L | Rm,Rn | Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits | 0011nnnnmmmm0101 | 2 to 5* | — |
| DT | Rn | Rn - 1 → Rn, if Rn = 0, 1 → T, else 0 → T | 0100nnnn00010000 | 1 | Comparison result |
| EXTS.B | Rm,Rn | A byte in Rm is sign-extended → Rn | 0110nnnnmmmm1110 | 1 | — |
| EXTS.W | Rm,Rn | A word in Rm is sign-extended → Rn | 0110nnnnmmmm1111 | 1 | — |
| EXTU.B | Rm,Rn | A byte in Rm is zero-extended → Rn | 0110nnnnmmmm1100 | 1 | — |
| EXTU.W | Rm,Rn | A word in Rm is zero-extended → Rn | 0110nnnnmmmm1101 | 1 | — |
| MAC.L | @Rm+,@Rn+ | Signed operation of (Rn) × (Rm) + MAC → MAC, 32 × 32 + 64 → 64 bits | 0000nnnnmmmm1111 | 2 to 5* | — |
| MAC.W | @Rm+,@Rn+ | Signed operation of (Rn) × (Rm) + MAC → MAC, 16 × 16 + 64 → 64 bits | 0100nnnnmmmm1111 | 2 to 4* | — |
| MUL.L | Rm,Rn | Rn × Rm → MACL 32 × 32 → 32 bits | 0000nnnnmmmm0111 | 2 to 5* | — |
| MULS.W | Rm,Rn | Signed operation of Rn × Rm → MAC 16 × 16 → 32 bits | 0010nnnnmmmm1111 | 1 to 3* | — |
| MULU.W | Rm,Rn | Unsigned operation of Rn × Rm → MAC 16 × 16 → 32 bits | 0010nnnnmmmm1110 | 1 to 3* | — |
| NEG | Rm,Rn | 0-Rm → Rn | 0110nnnnmmmm1011 | 1 | — |
| NEGC | Rm,Rn | 0-Rm-T → Rn, Borrow → T | 0110nnnnmmmm1010 | 1 | Borrow |
| SUB | Rm,Rn | Rn-Rm → Rn | 0011nnnnmmmm1000 | 1 | — |
| SUBC | Rm,Rn | Rn-Rm–T → Rn, Borrow → T | 0011nnnnmmmm1010 | 1 | Borrow |
| SUBV | Rm,Rn | Rn-Rm → Rn, Underflow → T | 0011nnnnmmmm1011 | 1 | Overflow |

Note: * Indicates the number of execution cycles for normal operation.

### 2.5.4 Logic Operation Instructions

**Table 2.13 Logic Operation Instructions**

| Instruction | Operation | Code | Execution Cycles | T Bit |
|---|---|---|---|---|
| AND Rm,Rn | Rn & Rm → Rn | 0010nnnnmmmm1001 | 1 | — |
| AND #imm,R0 | R0 & imm → R0 | 11001001iiiiiiii | 1 | — |
| AND.B #imm,@(R0,GBR) | (R0 + GBR) & imm → (R0 + GBR) | 11001101iiiiiiii | 3 | — |
| NOT Rm,Rn | ~Rm → Rn | 0110nnnnmmmm0111 | 1 | — |
| OR Rm,Rn | Rn \| Rm → Rn | 0010nnnnmmmm1011 | 1 | — |
| OR #imm,R0 | R0 \| imm → R0 | 11001011iiiiiiii | 1 | — |
| OR.B #imm,@(R0,GBR) | (R0 + GBR) \| imm → (R0 + GBR) | 11001111iiiiiiii | 3 | — |
| TAS.B @Rn | If (Rn) is 0, 1 → T; 1 → MSB of (Rn) | 0100nnnn00011011 | 4 | Test result |
| TST Rm,Rn | Rn & Rm; if the result is 0, 1 → T | 0010nnnnmmmm1000 | 1 | Test result |
| TST #imm,R0 | R0 & imm; if the result is 0, 1 → T | 11001000iiiiiiii | 1 | Test result |
| TST.B #imm,@(R0,GBR) | (R0 + GBR) & imm; if the result is 0, 1 → T | 11001100iiiiiiii | 3 | Test result |
| XOR Rm,Rn | Rn ^ Rm → Rn | 0010nnnnmmmm1010 | 1 | — |
| XOR #imm,R0 | R0 ^ imm → R0 | 11001010iiiiiiii | 1 | — |
| XOR.B #imm,@(R0,GBR) | (R0 + GBR) ^ imm → (R0 + GBR) | 11001110iiiiiiii | 3 | — |

### 2.5.5 Shift Instructions

**Table 2.14 Shift Instructions**

| Instruction | Operation | Code | Execution Cycles | T Bit |
|---|---|---|---|---|
| ROTL Rn | T ← Rn ← MSB | 0100nnnn00000100 | 1 | MSB |
| ROTR Rn | LSB → Rn → T | 0100nnnn00000101 | 1 | LSB |
| ROTCL Rn | T ← Rn ← T | 0100nnnn00100100 | 1 | MSB |
| ROTCR Rn | T → Rn → T | 0100nnnn00100101 | 1 | LSB |
| SHAL Rn | T ← Rn ← 0 | 0100nnnn00100000 | 1 | MSB |
| SHAR Rn | MSB → Rn → T | 0100nnnn00100001 | 1 | LSB |
| SHLL Rn | T ← Rn ← 0 | 0100nnnn00000000 | 1 | MSB |
| SHLR Rn | 0 → Rn → T | 0100nnnn00000001 | 1 | LSB |
| SHLL2 Rn | Rn << 2 → Rn | 0100nnnn00001000 | 1 | — |
| SHLR2 Rn | Rn >> 2 → Rn | 0100nnnn00001001 | 1 | — |
| SHLL8 Rn | Rn << 8 → Rn | 0100nnnn00011000 | 1 | — |
| SHLR8 Rn | Rn >> 8 → Rn | 0100nnnn00011001 | 1 | — |
| SHLL16 Rn | Rn << 16 → Rn | 0100nnnn00101000 | 1 | — |
| SHLR16 Rn | Rn >> 16 → Rn | 0100nnnn00101001 | 1 | — |

### 2.5.6 Branch Instructions

**Table 2.15 Branch Instructions**

| Instruction | | Operation | Code | Execution Cycles | T Bit |
|---|---|---|---|---|---|
| BF | label | If T = 0, disp × 2 + PC → PC;<br>if T = 1, nop | 10001011dddddddd | 3/1* | — |
| BF/S | label | Delayed branch, if T = 0, disp × 2 + PC → PC;<br>if T = 1, nop | 10001111dddddddd | 2/1* | — |
| BT | label | If T = 1, disp × 2 + PC → PC;<br>if T = 0, nop | 10001001dddddddd | 3/1* | — |
| BT/S | label | Delayed branch, if T = 1, disp × 2 + PC → PC;<br>if T = 0, nop | 10001101dddddddd | 2/1* | — |
| BRA | label | Delayed branch,<br>disp × 2 + PC → PC | 1010dddddddddddd | 2 | — |
| BRAF | Rm | Delayed branch,<br>Rm + PC → PC | 0000mmmm00100011 | 2 | — |
| BSR | label | Delayed branch, PC → PR,<br>disp × 2 + PC → PC | 1011dddddddddddd | 2 | — |
| BSRF | Rm | Delayed branch, PC → PR,<br>Rm + PC → PC | 0000mmmm00000011 | 2 | — |
| JMP | @Rm | Delayed branch, Rm → PC | 0100mmmm00101011 | 2 | — |
| JSR | @Rm | Delayed branch, PC → PR,<br>Rm → PC | 0100mmmm00001011 | 2 | — |
| RTS | | Delayed branch, PR → PC | 0000000000001011 | 2 | — |

Note: * One cycle when the branch is not executed.

### 2.5.7 System Control Instructions

**Table 2.16 System Control Instructions**

| Instruction | Operation | Code | Execution Cycles | T Bit |
|---|---|---|---|---|
| CLRT | 0 → T | 0000000000001000 | 1 | 0 |
| CLRMAC | 0 → MACH, MACL | 0000000000101000 | 1 | — |
| LDC Rm,SR | Rm → SR | 0100mmmm00001110 | 6 | LSB |
| LDC Rm,GBR | Rm → GBR | 0100mmmm00011110 | 4 | — |
| LDC Rm,VBR | Rm → VBR | 0100mmmm00101110 | 4 | — |
| LDC.L @Rm+,SR | (Rm) → SR, Rm + 4 → Rm | 0100mmmm00000111 | 8 | LSB |
| LDC.L @Rm+,GBR | (Rm) → GBR, Rm + 4 → Rm | 0100mmmm00010111 | 4 | — |
| LDC.L @Rm+,VBR | (Rm) → VBR, Rm + 4 → Rm | 0100mmmm00100111 | 4 | — |
| LDS Rm,MACH | Rm → MACH | 0100mmmm00001010 | 1 | — |
| LDS Rm,MACL | Rm → MACL | 0100mmmm00011010 | 1 | — |
| LDS Rm,PR | Rm → PR | 0100mmmm00101010 | 1 | — |
| LDS.L @Rm+,MACH | (Rm) → MACH, Rm + 4 → Rm | 0100mmmm00000110 | 1 | — |
| LDS.L @Rm+,MACL | (Rm) → MACL, Rm + 4 → Rm | 0100mmmm00010110 | 1 | — |
| LDS.L @Rm+,PR | (Rm) → PR, Rm + 4 → Rm | 0100mmmm00100110 | 1 | — |
| NOP | No operation | 0000000000001001 | 1 | — |
| RTE | Delayed branch, Stack area → PC/SR | 0000000000101011 | 5 | — |
| SETT | 1 → T | 0000000000011000 | 1 | 1 |
| SLEEP | Sleep | 0000000000011011 | 4* | — |
| STC SR,Rn | SR → Rn | 0000nnnn00000010 | 1 | — |
| STC GBR,Rn | GBR → Rn | 0000nnnn00010010 | 1 | — |
| STC VBR,Rn | VBR → Rn | 0000nnnn00100010 | 1 | — |
| STC.L SR,@–Rn | Rn–4 → Rn, SR → (Rn) | 0100nnnn00000011 | 1 | — |
| STC.L GBR,@–Rn | Rn–4 → Rn, GBR → (Rn) | 0100nnnn00010011 | 1 | — |
| STC.L VBR,@–Rn | Rn–4 → Rn, VBR → (Rn) | 0100nnnn00100011 | 1 | — |

| Instruction | Operation | Code | Execution Cycles | T Bit |
|---|---|---|---|---|
| STS MACH,Rn | MACH → Rn | 0000nnnn00001010 | 1 | — |
| STS MACL,Rn | MACL → Rn | 0000nnnn00011010 | 1 | — |
| STS PR,Rn | PR → Rn | 0000nnnn00101010 | 1 | — |
| STS.L MACH,@–Rn | Rn–4 → Rn, MACH → (Rn) | 0100nnnn00000010 | 1 | — |
| STS.L MACL,@–Rn | Rn–4 → Rn, MACL → (Rn) | 0100nnnn00010010 | 1 | — |
| STS.L PR,@–Rn | Rn–4 → Rn, PR → (Rn) | 0100nnnn00100010 | 1 | — |
| TRAPA #imm | PC/SR → Stack area,<br>(imm × 4 + VBR) → PC | 11000011iiiiiiii | 8 | — |

Note: * Number of execution cycles until this LSI enters sleep mode.

About the number of execution cycles:

The table lists the minimum number of execution cycles. In practice, the number of execution cycles will be increased depending on the conditions such as:

- When there is a conflict between instruction fetch and data access
- When the destination register of a load instruction (memory → register) is also used by the instruction immediately after the load instruction.

## 2.6 Processing States

The CPU has the five processing states: reset, exception handling, bus release, program execution, and power-down. Figure 2.4 shows the CPU state transition.

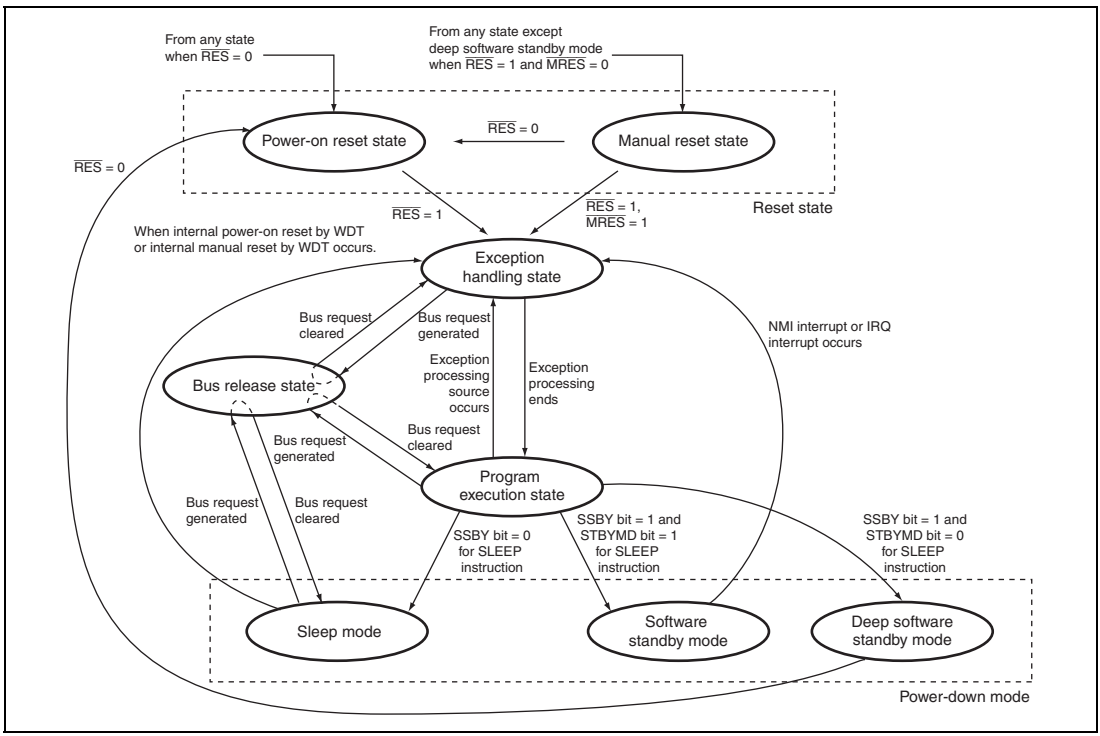



**Figure 2.4 Transitions between Processing States**

- Reset state

  The CPU is reset. When the $\overline{\text{RES}}$ pin is low, the CPU enters the power-on reset state. When the $\overline{\text{RES}}$ pin is high and $\overline{\text{MRES}}$ pin is low, the CPU enters the manual reset state.

- Exception handling state

  This state is a transitional state in which the CPU processing state changes due to a request for exception handling such as a reset or an interrupt.

  When a reset occurs, the execution start address as the initial value of the program counter (PC) and the initial value of the stack pointer (SP) are fetched from the exception handling vector table. Then, a branch is made for the start address to execute a program.

  When an interrupt occurs, the PC and status register (SR) are saved in the stack area pointed to by SP. The start address of an exception handling routine is fetched from the exception handling vector table and a branch to the address is made to execute a program.

  Then the processing state enters the program execution state.

- Program execution state

  The CPU executes programs sequentially.

- Power-down state

  The CPU stops to reduce power consumption. The SLEEP instruction makes the CPU enter sleep mode, software standby mode, or deep software standby mode.

- Bus release state

  In the bus release state, the CPU releases access rights to the bus to the device that has requested them.

# Section 3 MCU Operating Modes

## 3.1 Selection of Operating Modes

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

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

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

**Table 3.1 Selection of Operating Modes**

| | Pin Setting | | | | | Bus Width of CS0 Space | |
|---|---|---|---|---|---|---|---|
| **Mode No.** | **FWE** | **MD1** | **MD0*[1]** | **Mode Name** | **On-Chip ROM** | **SH7131/SH7136** | **SH7132/SH7137** |
| Mode 0 | 0 | 0 | 0 | MCU extension mode 0 | Disabled | — | 8 |
| Mode 2 | 0 | 1 | 0 | MCU extension mode 2 | Enabled | — | 8 |
| Mode 3 | 0 | 1 | 1 | Single chip mode | Enabled | — | — |
| Mode 4*[2] | 1 | 0 | 0 | Boot mode | Enabled | — | — |
| Mode 5*[2] | 1 | 0 | 1 | User boot mode | Enabled | — | 8 |
| Mode 6*[2] | 1 | 1 | 0 | User programming mode | Enabled | — | 8 |
| Mode 7*[2] | 1 | 1 | 1 | | | — | — |

Notes: 1. The SH7131 and SH7136 do not have the MD0 pin and only supports the following operating modes according to the combination of the FWE and MD1 pins.
Single chip mode: FWE pin = 0 and MD1 pin = 1
Boot mode: FWE pin = 1 and MD1 pin = 0
User programming mode: FWE pin= 1 and MD1 pin = 1
2. Flash memory programming mode.

## 3.2 Input/Output Pins

Table 3.2 describes the configuration of operating mode related pin.

**Table 3.2 Pin Configuration**

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

## 3.3 Operating Modes

### 3.3.1 Mode 0 (MCU Extension Mode 0)

CS0 space becomes external memory spaces with 8-bit bus width in SH7132 and SH7137.

### 3.3.2 Mode 2 (MCU Extension Mode 2)

The on-chip ROM is enabled and CS space can be used in this mode.

### 3.3.3 Mode 3 (Single Chip Mode)

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

## 3.4 Address Map

The address maps for the operating modes are shown in figures 3.1 to 3.4.

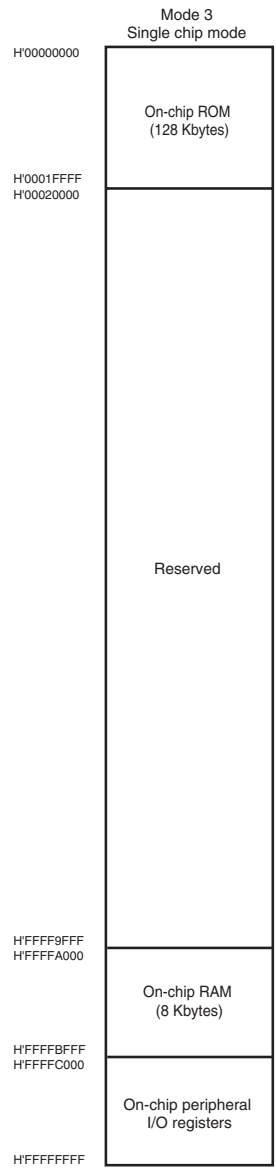



**Figure 3.1 Address Map for Each Operating Mode in SH7131 (128-Kbyte Flash Memory Version)**

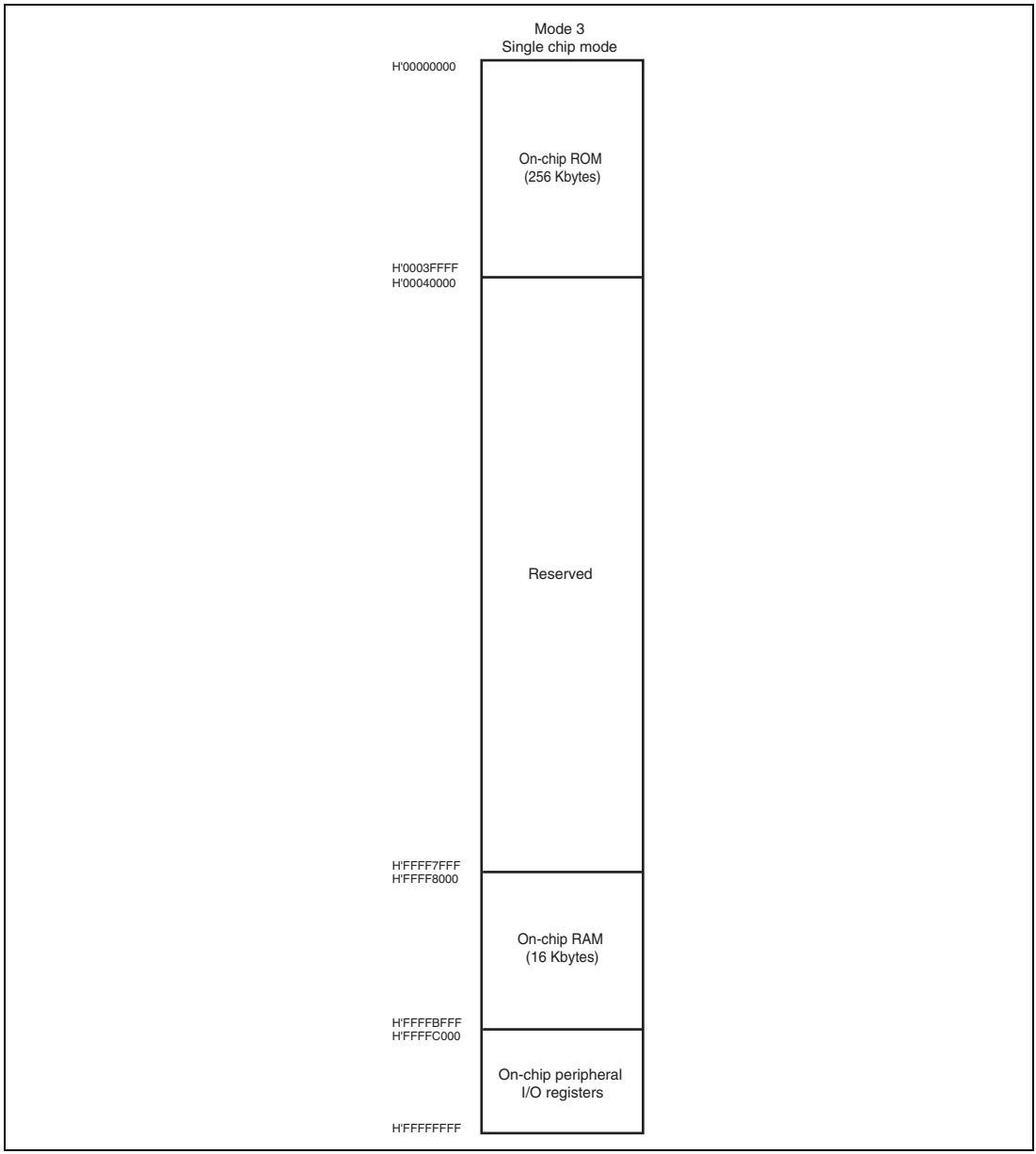



**Figure 3.2 Address Map for Each Operating Mode in SH7131 and SH7136 (256-Kbyte Flash Memory Version)**

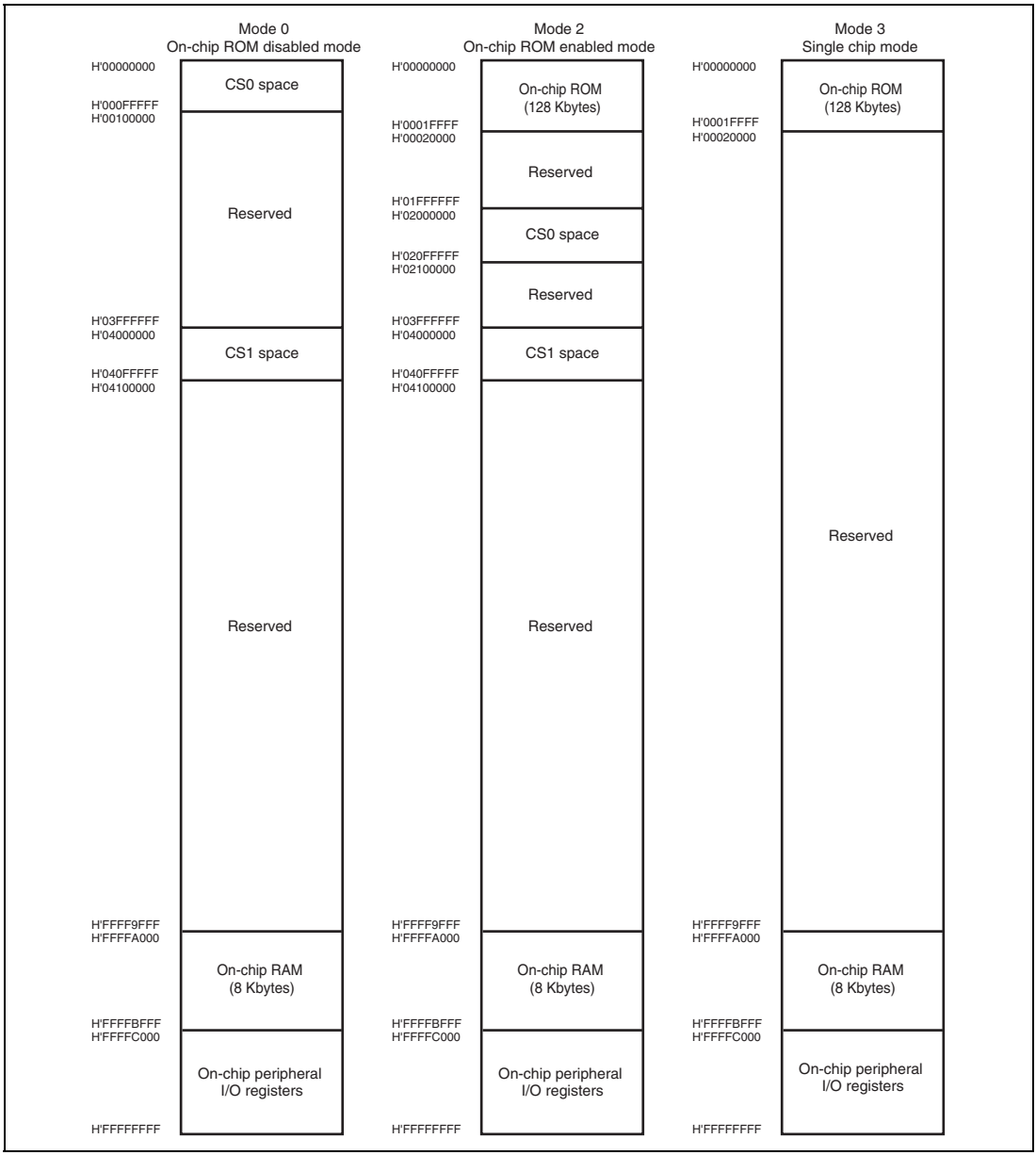



**Figure 3.3 Address Map for Each Operating Mode in SH7132 (128-Kbyte Flash Memory Version)**

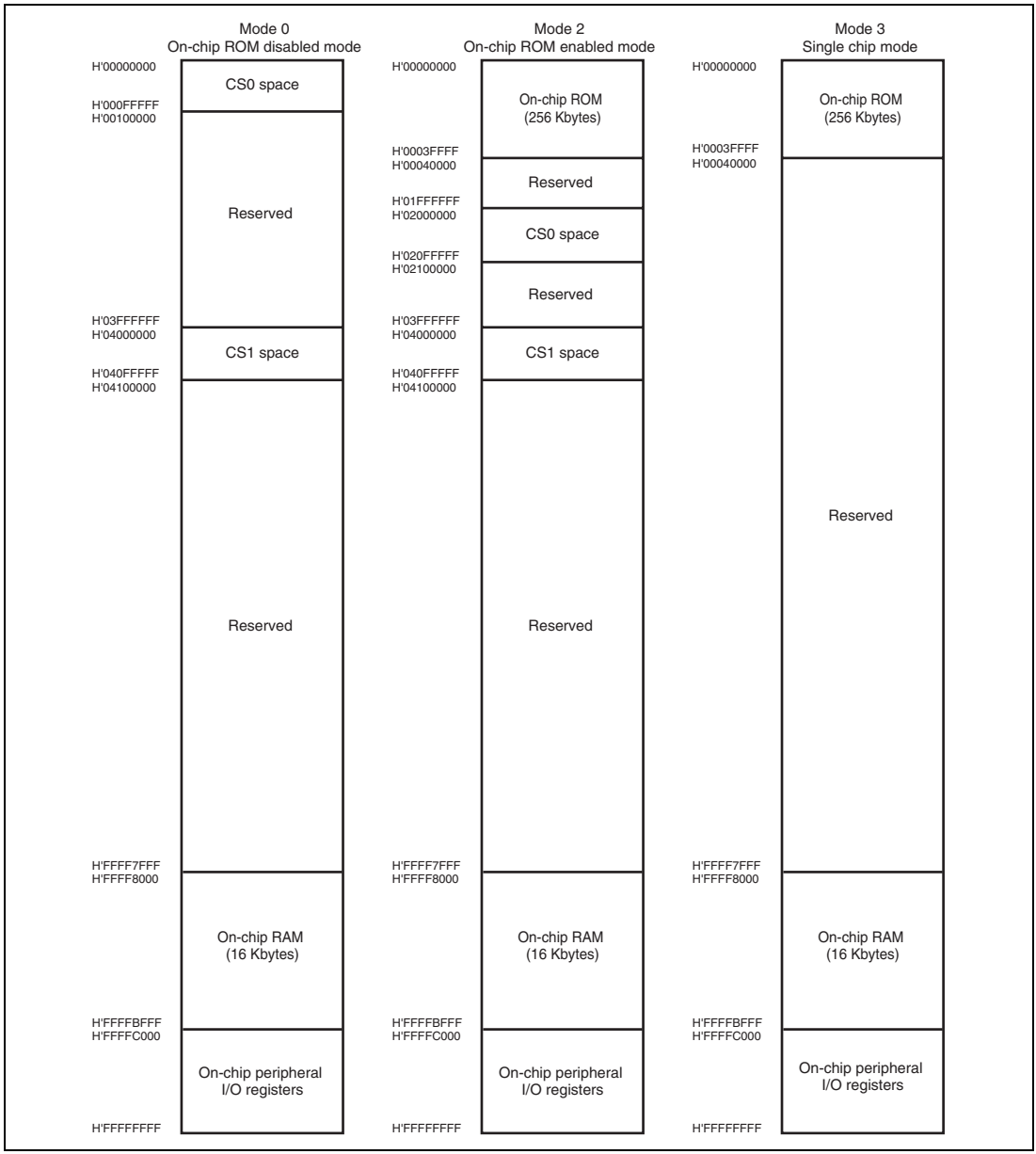


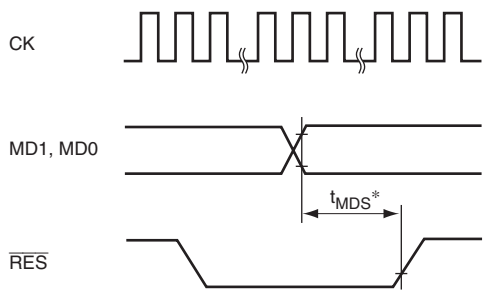

**Figure 3.4 Address Map for Each Operating Mode in SH7132 and SH7137 (256-Kbyte Flash Memory Version)**

## 3.5 Initial State in This LSI

In the initial state of this LSI, some of on-chip modules are set in module standby state for saving power. When operating these modules, clear module standby state according to the procedure in section 24, Power-Down Modes.

## 3.6 Note on Changing Operating Mode

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




Note: * See section 26.3.2, Control Signal Timing.

**Figure 3.5 Reset Input Timing when Changing Operating Mode**

# Section 4 Clock Pulse Generator (CPG)

This LSI has a clock pulse generator (CPG) that generates an internal clock (Iϕ), a bus clock (Bϕ), a peripheral clock (Pϕ), and clocks (MIϕ and MPϕ) for the MTU2S and MTU2 modules. The CPG also controls power-down modes.

## 4.1 Features

- Five clocks generated independently

  An internal clock (Iϕ) for the CPU; a peripheral clock (Pϕ) for the on-chip peripheral modules; a bus clock (Bϕ = CK) for the external bus interface; a MTU2S clock (MIϕ) for the on-chip MTU2S module; and a MTU2 clock (MPϕ) for the on-chip MTU2 module.
- Frequency change function

  Frequencies of the internal clock (Iϕ), bus clock (Bϕ), peripheral clock (Pϕ), MTU2S clock (MIϕ), and MTU2 clock (MPϕ) can be changed independently using the divider circuit within the CPG. Frequencies are changed by software using the frequency control register (FRQCR) setting.
- Power-down mode control

  The clock can be stopped in sleep mode and standby mode and specific modules can be stopped using the module standby function.
- Oscillation stop detection

  If the clock supplied through the clock input pin stops for any reason, the timer pins can be automatically placed in the high-impedance state.

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

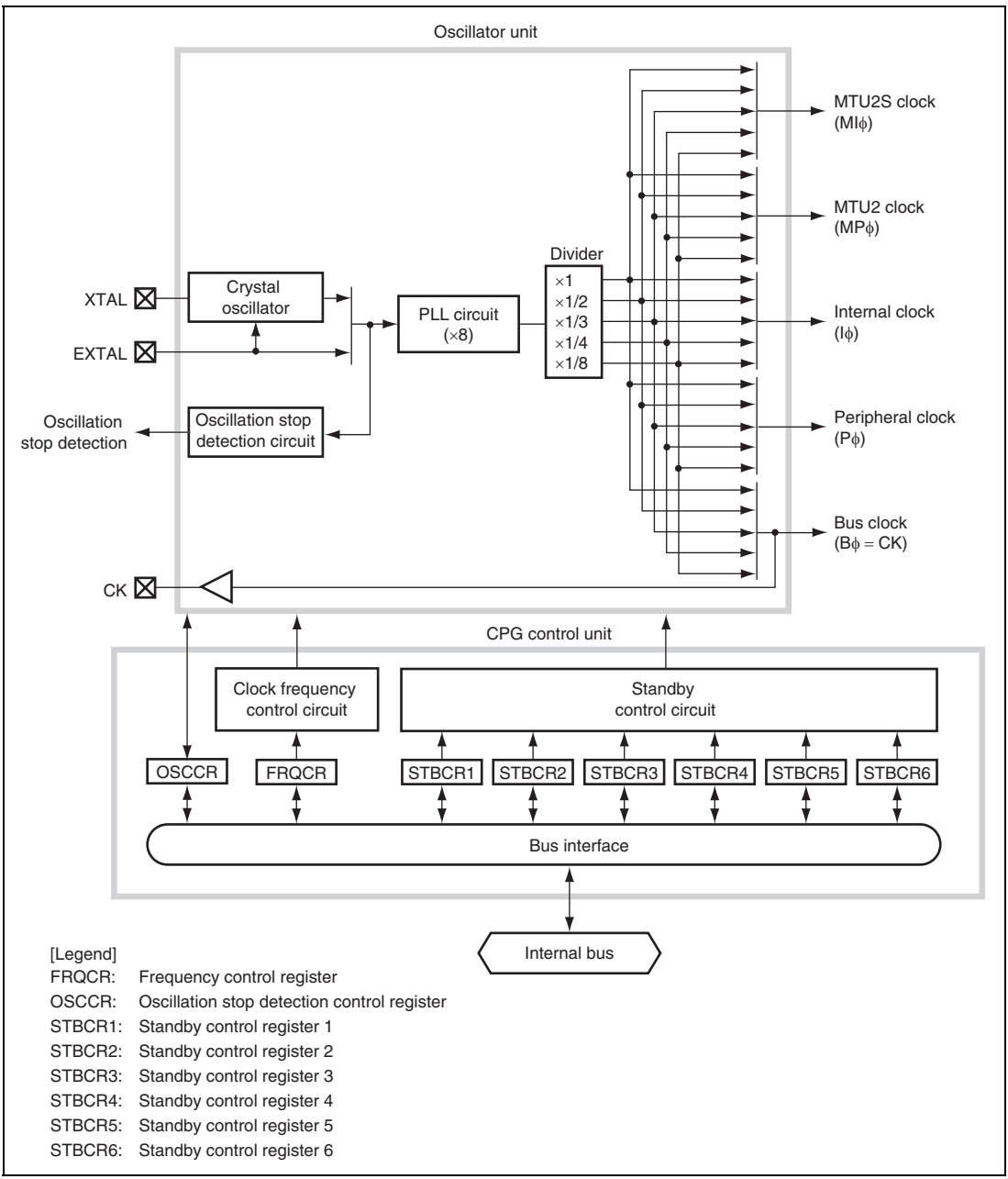



**Figure 4.1 Block Diagram of Clock Pulse Generator**

The clock pulse generator blocks function as follows:

**PLL Circuit:** The PLL circuit multiples the clock frequency input from the crystal oscillator or the EXTAL pin by 8. The multiplication ratio is fixed at ×8.

**Crystal Oscillator:** The crystal oscillator is an oscillator circuit when a crystal resonator is connected to the XTAL and EXTAL pins.

**Divider:** The divider generates clocks with the frequencies to be used by the internal clock (Iϕ), bus clock (Bϕ), peripheral clock (Pϕ), MTU2S clock (MIϕ), and MTU2 clock (MPϕ).

The frequencies can be selected from 1, 1/2, 1/3, 1/4, and 1/8 times the frequency output from the PLL circuit. The division ratio should be specified in the frequency control register (FRQCR).

**Oscillation Stop Detection Circuit:** This circuit detects an abnormal condition in the crystal oscillator.

**Clock Frequency Control Circuit:** The clock frequency control circuit controls the clock frequency according to the setting in the frequency control register (FRQCR).

**Standby Control Circuit:** The standby control circuit controls the state of the on-chip oscillator circuit and other modules in sleep or standby mode.

**Frequency Control Register (FRQCR):** The frequency control register (FRQCR) has control bits for the frequency division ratios of the internal clock (Iϕ), bus clock (Bϕ), peripheral clock (Pϕ), MTU2S clock (MIϕ), and MTU2 clock (MPϕ).

**Oscillation Stop Detection Control Register (OSCCR):** The oscillation stop detection control register (OSCCR) has an oscillation stop detection flag and a bit for selecting flag status output through an external pin.

**Standby Control Registers 1 to 6 (STBCR1 to STBCR6):** The standby control register (STBCR) has bits for controlling the power-down modes. For details, see section 24, Power-Down Modes.

Table 4.1 shows the operating clock for each module.

**Table 4.1 Operating Clock for Each Module**

| Operating Clock | Operating Module | Operating Clock | Operating Module |
|---|---|---|---|
| Internal clock (Iϕ) | CPU | Peripheral clock (Pϕ) | POE |
| | UBC* | | SCI |
| | ROM | | SSU |
| | RAM | | RCAN-ET |
| | | | $I^2C$ |
| | | | A/D |
| | | | CMT |
| | | | WDT |
| Bus clock (Bϕ) | BSC | MTU2 clock (MPϕ) | MTU2 |
| | DTC | MTU2S clock (MIϕ) | MTU2S |

Note: * SH7136 and SH7137 only.

## 4.2 Input/Output Pins

Table 4.2 shows the CPG pin configuration.

**Table 4.2 Pin Configuration**

| Pin Name | Abbr. | I/O | Description |
|---|---|---|---|
| Crystal input/output pins (clock input pins) | XTAL | Output | Connects a crystal resonator. |
| | EXTAL | Input | Connects a crystal resonator or an external clock. |
| Clock output pin | CK | Output | Outputs an external clock. |

Note: To use the clock output (CK) pin, appropriate settings may be needed for the pin in the pin function controller (PFC) in some cases. For details, refer to section 20, Pin Function Controller (PFC).

## 4.3 Clock Operating Mode

Table 4.3 shows the clock operating mode of this LSI.

**Table 4.3 Clock Operating Mode**

| Clock Operating Mode | Clock I/O | | PLL Circuit | Input to Divider |
|---|---|---|---|---|
| | Source | Output | | |
| 1 | EXTAL input or crystal resonator | CK* | ON (×8) | ×8 |

Note: * To output the clock through the clock output (CK) pin, appropriate settings should be made in the pin function controller (PFC). For details, refer to section 20, Pin Function Controller (PFC).

**Mode 1:** The frequency of the external clock input from the EXTAL pin is multiplied by 8 in the PLL circuit before being supplied to the on-chip modules in this LSI, which eliminates the need to generate a high-frequency clock outside the LSI. Since the input clock frequency ranging from 5 MHz to 12.5 MHz can be used, the internal clock (Iϕ) frequency ranges from 10 MHz to 80 MHz.

Maximum operating frequencies:
Iϕ = 80 MHz, Bϕ = 40 MHz, Pϕ = 40 MHz, MIϕ = 80 MHz, and MPϕ = 40 MHz

Table 4.4 shows the frequency division ratios that can be specified with FRQCR.

## Table 4.4 Frequency Division Ratios Specifiable with FRQCR

| PLL Multiplication Ratio | FRQCR Division Ratio Setting | | | | | Clock Ratio | | | | | Clock Frequency (MHz)* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Iϕ | Bϕ | Pϕ | MIϕ | MPϕ | Iϕ | Bϕ | Pϕ | MIϕ | MPϕ | Input Clock | Iϕ | Bϕ | Pϕ | MIϕ | MPϕ |
| ×8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1 | 1 | 1 | 1 | 1 | 10 | 10 | 10 | 10 | 10 | 10 |
| | 1/4 | 1/8 | 1/8 | 1/8 | 1/8 | 2 | 1 | 1 | 1 | 1 | | 20 | 10 | 10 | 10 | 10 |
| | 1/4 | 1/8 | 1/8 | 1/4 | 1/8 | 2 | 1 | 1 | 2 | 1 | | 20 | 10 | 10 | 20 | 10 |
| | 1/4 | 1/4 | 1/8 | 1/8 | 1/8 | 2 | 2 | 1 | 1 | 1 | | 20 | 20 | 10 | 10 | 10 |
| | 1/4 | 1/4 | 1/8 | 1/4 | 1/8 | 2 | 2 | 1 | 2 | 1 | | 20 | 20 | 10 | 20 | 10 |
| | 1/4 | 1/4 | 1/8 | 1/4 | 1/4 | 2 | 2 | 1 | 2 | 2 | | 20 | 20 | 10 | 20 | 20 |
| | 1/4 | 1/4 | 1/4 | 1/4 | 1/4 | 2 | 2 | 2 | 2 | 2 | | 20 | 20 | 20 | 20 | 20 |
| | 1/3 | 1/3 | 1/3 | 1/3 | 1/3 | 8/3 | 8/3 | 8/3 | 8/3 | 8/3 | | 26 | 26 | 26 | 26 | 26 |
| | 1/2 | 1/8 | 1/8 | 1/8 | 1/8 | 4 | 1 | 1 | 1 | 1 | | 40 | 10 | 10 | 10 | 10 |
| | 1/2 | 1/8 | 1/8 | 1/4 | 1/8 | 4 | 1 | 1 | 2 | 1 | | 40 | 10 | 10 | 20 | 10 |
| | 1/2 | 1/8 | 1/8 | 1/2 | 1/8 | 4 | 1 | 1 | 4 | 1 | | 40 | 10 | 10 | 40 | 10 |
| | 1/2 | 1/4 | 1/8 | 1/8 | 1/8 | 4 | 2 | 1 | 1 | 1 | | 40 | 20 | 10 | 10 | 10 |
| | 1/2 | 1/4 | 1/8 | 1/4 | 1/8 | 4 | 2 | 1 | 2 | 1 | | 40 | 20 | 10 | 20 | 10 |
| | 1/2 | 1/4 | 1/8 | 1/4 | 1/4 | 4 | 2 | 1 | 2 | 2 | | 40 | 20 | 10 | 20 | 20 |
| | 1/2 | 1/4 | 1/8 | 1/2 | 1/8 | 4 | 2 | 1 | 4 | 1 | | 40 | 20 | 10 | 40 | 10 |
| | 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 4 | 2 | 1 | 4 | 2 | | 40 | 20 | 10 | 40 | 20 |
| | 1/2 | 1/4 | 1/4 | 1/4 | 1/4 | 4 | 2 | 2 | 2 | 2 | | 40 | 20 | 20 | 20 | 20 |
| | 1/2 | 1/4 | 1/4 | 1/2 | 1/4 | 4 | 2 | 2 | 4 | 2 | | 40 | 20 | 20 | 40 | 20 |
| | 1/2 | 1/2 | 1/8 | 1/8 | 1/8 | 4 | 4 | 1 | 1 | 1 | | 40 | 40 | 10 | 10 | 10 |
| | 1/2 | 1/2 | 1/8 | 1/4 | 1/8 | 4 | 4 | 1 | 2 | 1 | | 40 | 40 | 10 | 20 | 10 |
| | 1/2 | 1/2 | 1/8 | 1/4 | 1/4 | 4 | 4 | 1 | 2 | 2 | | 40 | 40 | 10 | 20 | 20 |
| | 1/2 | 1/2 | 1/8 | 1/2 | 1/8 | 4 | 4 | 1 | 4 | 1 | | 40 | 40 | 10 | 40 | 10 |
| | 1/2 | 1/2 | 1/8 | 1/2 | 1/4 | 4 | 4 | 1 | 4 | 2 | | 40 | 40 | 10 | 40 | 20 |
| | 1/2 | 1/2 | 1/8 | 1/2 | 1/2 | 4 | 4 | 1 | 4 | 4 | | 40 | 40 | 10 | 40 | 40 |
| | 1/2 | 1/2 | 1/4 | 1/4 | 1/4 | 4 | 4 | 2 | 2 | 2 | | 40 | 40 | 20 | 20 | 20 |
| | 1/2 | 1/2 | 1/4 | 1/2 | 1/4 | 4 | 4 | 2 | 4 | 2 | | 40 | 40 | 20 | 40 | 20 |
| | 1/2 | 1/2 | 1/4 | 1/2 | 1/2 | 4 | 4 | 2 | 4 | 4 | | 40 | 40 | 20 | 40 | 40 |

| PLL Multiplication Ratio | FRQCR Division Ratio Setting | | | | | Clock Ratio | | | | | Clock Frequency (MHz)* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Iϕ | Bϕ | Pϕ | MIϕ | MPϕ | Iϕ | Bϕ | Pϕ | MIϕ | MPϕ | Input Clock | Iϕ | Bϕ | Pϕ | MIϕ | MPϕ |
| ×8 | 1/2 | 1/2 | 1/2 | 1/2 | 1/2 | 4 | 4 | 4 | 4 | 4 | 10 | 40 | 40 | 40 | 40 | 40 |
| | 1/1 | 1/8 | 1/8 | 1/8 | 1/8 | 8 | 1 | 1 | 1 | 1 | | 80 | 10 | 10 | 10 | 10 |
| | 1/1 | 1/8 | 1/8 | 1/4 | 1/8 | 8 | 1 | 1 | 2 | 1 | | 80 | 10 | 10 | 20 | 10 |
| | 1/1 | 1/8 | 1/8 | 1/2 | 1/8 | 8 | 1 | 1 | 4 | 1 | | 80 | 10 | 10 | 40 | 10 |
| | 1/1 | 1/8 | 1/8 | 1/1 | 1/8 | 8 | 1 | 1 | 8 | 1 | | 80 | 10 | 10 | 80 | 10 |
| | 1/1 | 1/4 | 1/8 | 1/8 | 1/8 | 8 | 2 | 1 | 1 | 1 | | 80 | 20 | 10 | 10 | 10 |
| | 1/1 | 1/4 | 1/8 | 1/4 | 1/8 | 8 | 2 | 1 | 2 | 1 | | 80 | 20 | 10 | 20 | 10 |
| | 1/1 | 1/4 | 1/8 | 1/4 | 1/4 | 8 | 2 | 1 | 2 | 2 | | 80 | 20 | 10 | 20 | 20 |
| | 1/1 | 1/4 | 1/8 | 1/2 | 1/8 | 8 | 2 | 1 | 4 | 1 | | 80 | 20 | 10 | 40 | 10 |
| | 1/1 | 1/4 | 1/8 | 1/2 | 1/4 | 8 | 2 | 1 | 4 | 2 | | 80 | 20 | 10 | 40 | 20 |
| | 1/1 | 1/4 | 1/8 | 1/1 | 1/8 | 8 | 2 | 1 | 8 | 1 | | 80 | 20 | 10 | 80 | 10 |
| | 1/1 | 1/4 | 1/8 | 1/1 | 1/4 | 8 | 2 | 1 | 8 | 2 | | 80 | 20 | 10 | 80 | 20 |
| | 1/1 | 1/4 | 1/4 | 1/4 | 1/4 | 8 | 2 | 2 | 2 | 2 | | 80 | 20 | 20 | 20 | 20 |
| | 1/1 | 1/4 | 1/4 | 1/2 | 1/4 | 8 | 2 | 2 | 4 | 2 | | 80 | 20 | 20 | 40 | 20 |
| | 1/1 | 1/4 | 1/4 | 1/1 | 1/4 | 8 | 2 | 2 | 8 | 2 | | 80 | 20 | 20 | 80 | 20 |
| | 1/1 | 1/3 | 1/3 | 1/3 | 1/3 | 8 | 8/3 | 8/3 | 8/3 | 8/3 | | 80 | 26 | 26 | 26 | 26 |
| | 1/1 | 1/3 | 1/3 | 1/1 | 1/3 | 8 | 8/3 | 8/3 | 8 | 8/3 | | 80 | 26 | 26 | 80 | 26 |
| | 1/1 | 1/2 | 1/8 | 1/8 | 1/8 | 8 | 4 | 1 | 1 | 1 | | 80 | 40 | 10 | 10 | 10 |
| | 1/1 | 1/2 | 1/8 | 1/4 | 1/8 | 8 | 4 | 1 | 2 | 1 | | 80 | 40 | 10 | 20 | 10 |
| | 1/1 | 1/2 | 1/8 | 1/4 | 1/4 | 8 | 4 | 1 | 2 | 2 | | 80 | 40 | 10 | 20 | 20 |
| | 1/1 | 1/2 | 1/8 | 1/2 | 1/8 | 8 | 4 | 1 | 4 | 1 | | 80 | 40 | 10 | 40 | 10 |
| | 1/1 | 1/2 | 1/8 | 1/2 | 1/4 | 8 | 4 | 1 | 4 | 2 | | 80 | 40 | 10 | 40 | 20 |
| | 1/1 | 1/2 | 1/8 | 1/2 | 1/2 | 8 | 4 | 1 | 4 | 4 | | 80 | 40 | 10 | 40 | 40 |
| | 1/1 | 1/2 | 1/8 | 1/1 | 1/8 | 8 | 4 | 1 | 8 | 1 | | 80 | 40 | 10 | 80 | 10 |
| | 1/1 | 1/2 | 1/8 | 1/1 | 1/4 | 8 | 4 | 1 | 8 | 2 | | 80 | 40 | 10 | 80 | 20 |
| | 1/1 | 1/2 | 1/8 | 1/1 | 1/2 | 8 | 4 | 1 | 8 | 4 | | 80 | 40 | 10 | 80 | 40 |
| | 1/1 | 1/2 | 1/4 | 1/4 | 1/4 | 8 | 4 | 2 | 2 | 2 | | 80 | 40 | 20 | 20 | 20 |
| | 1/1 | 1/2 | 1/4 | 1/2 | 1/4 | 8 | 4 | 2 | 4 | 2 | | 80 | 40 | 20 | 40 | 20 |

| PLL Multiplication Ratio | FRQCR Division Ratio Setting | | | | | Clock Ratio | | | | | Clock Frequency (MHz)* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Iϕ | Bϕ | Pϕ | MIϕ | MPϕ | Iϕ | Bϕ | Pϕ | MIϕ | MPϕ | Input Clock | Iϕ | Bϕ | Pϕ | MIϕ | MPϕ |
| ×8 | 1/1 | 1/2 | 1/4 | 1/2 | 1/2 | 8 | 4 | 2 | 4 | 4 | 10 | 80 | 40 | 20 | 40 | 40 |
| | 1/1 | 1/2 | 1/4 | 1/1 | 1/4 | 8 | 4 | 2 | 8 | 2 | | 80 | 40 | 20 | 80 | 20 |
| | 1/1 | 1/2 | 1/4 | 1/1 | 1/2 | 8 | 4 | 2 | 8 | 4 | | 80 | 40 | 20 | 80 | 40 |
| | 1/1 | 1/2 | 1/2 | 1/2 | 1/2 | 8 | 4 | 4 | 4 | 4 | | 80 | 40 | 40 | 40 | 40 |
| | 1/1 | 1/2 | 1/2 | 1/1 | 1/2 | 8 | 4 | 4 | 8 | 4 | | 80 | 40 | 40 | 80 | 40 |
| | 1/1 | 1/1 | 1/4 | 1/4 | 1/4 | 8 | 8 | 2 | 2 | 2 | 5 | 40 | 40 | 10 | 10 | 10 |
| | 1/1 | 1/1 | 1/4 | 1/2 | 1/4 | 8 | 8 | 2 | 4 | 2 | | 40 | 40 | 10 | 20 | 10 |
| | 1/1 | 1/1 | 1/4 | 1/2 | 1/2 | 8 | 8 | 2 | 4 | 4 | | 40 | 40 | 10 | 20 | 20 |
| | 1/1 | 1/1 | 1/4 | 1/1 | 1/4 | 8 | 8 | 2 | 8 | 2 | | 40 | 40 | 10 | 40 | 10 |
| | 1/1 | 1/1 | 1/4 | 1/1 | 1/2 | 8 | 8 | 2 | 8 | 4 | | 40 | 40 | 10 | 40 | 20 |
| | 1/1 | 1/1 | 1/4 | 1/1 | 1/1 | 8 | 8 | 2 | 8 | 8 | | 40 | 40 | 10 | 40 | 40 |
| | 1/1 | 1/1 | 1/3 | 1/3 | 1/3 | 8 | 8 | 8/3 | 8/3 | 8/3 | | 40 | 40 | 13 | 13 | 13 |
| | 1/1 | 1/1 | 1/3 | 1/1 | 1/3 | 8 | 8 | 8/3 | 8 | 8/3 | | 40 | 40 | 13 | 40 | 13 |
| | 1/1 | 1/1 | 1/3 | 1/1 | 1/1 | 8 | 8 | 8/3 | 8 | 8 | | 40 | 40 | 13 | 40 | 40 |
| | 1/1 | 1/1 | 1/2 | 1/2 | 1/2 | 8 | 8 | 4 | 4 | 4 | | 40 | 40 | 20 | 20 | 20 |
| | 1/1 | 1/1 | 1/2 | 1/1 | 1/2 | 8 | 8 | 4 | 8 | 4 | | 40 | 40 | 20 | 40 | 20 |
| | 1/1 | 1/1 | 1/2 | 1/1 | 1/1 | 8 | 8 | 4 | 8 | 8 | | 40 | 40 | 20 | 40 | 40 |
| | 1/1 | 1/1 | 1/1 | 1/1 | 1/1 | 8 | 8 | 8 | 8 | 8 | | 40 | 40 | 40 | 40 | 40 |

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

1. The PLL multiplication ratio is fixed at ×8. The division ratio can be selected from ×1, ×1/2, ×1/3, ×1/4, and ×1/8 for each clock by the setting in the frequency control register.
2. The output frequency of the PLL circuit is the product of the frequency of the input from the crystal resonator or EXTAL pin and the multiplication ratio (×8) of the PLL circuit.
3. The input to the divider is always the output from the PLL circuit.
4. The internal clock (Iϕ) frequency is the product of the frequency of the input from the crystal resonator or EXTAL pin, the multiplication ratio (×8) of the PLL circuit, and the division ratio of the divider. The resultant frequency must be a maximum of 80 MHz (maximum operating frequency).
5. The bus clock (Bϕ) frequency is the product of the frequency of the input from the crystal resonator or EXTAL pin, the multiplication ratio (×8) of the PLL circuit, and the division ratio of the divider. The resultant frequency must be a maximum of 40 MHz and equal to or lower than the internal clock (Iϕ) frequency.
6. The peripheral clock (Pϕ) frequency is the product of the frequency of the input from the crystal resonator or EXTAL pin, the multiplication ratio (×8) of the PLL circuit, and the division ratio of the divider. The resultant frequency must be a maximum of 40 MHz and equal to or lower than the bus clock (Bϕ) frequency.
7. When using the MTU2S and MTU2, the MTU2S clock (MIϕ) frequency must be equal to or lower than the internal clock (Iϕ) frequency and equal to or higher than the MTU2 clock (MPϕ) frequency. The MTU2 clock (MPϕ) frequency must be equal to or lower than the MTU2S clock (MIϕ) frequency and the bus clock (Bϕ) frequency, and equal to or higher than the peripheral clock frequency (Pϕ). The MTU2S clock (MIϕ) frequency and MTU2 clock (MPϕ) frequency are the product of the frequency of the input from the crystal resonator or EXTAL pin, the multiplication ratio (×8) of the PLL circuit, and the division ratio of the divider.
8. The frequency of the CK pin is always be equal to the bus clock (Bϕ) frequency.

## 4.4 Register Descriptions

The CPG has the following registers.

For details on the addresses of these registers and the states of these registers in each processing state, see section 25, List of Registers.

**Table 4.5 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Frequency control register | FRQCR | R/W | H'36DB | H'FFFFE800 | 16 |
| Oscillation stop detection control register | OSCCR | R/W | H'00 | H'FFFFE814 | 8 |

### 4.4.1 Frequency Control Register (FRQCR)

FRQCR is a 16-bit readable/writable register that specifies the frequency division ratios for the internal clock (Iϕ), bus clock (Bϕ), peripheral clock (Pϕ), MTU2S clock (MIϕ), and MTU2 clock (MPϕ). FRQCR can be accessed only in words.

FRQCR is initialized to H'36DB only by a power-on reset (except a power-on reset due to a WDT overflow).

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 to 12 | IFC[2:0] | 011 | R/W | Internal Clock (Iϕ) Frequency Division Ratio<br>Specify the division ratio of the internal clock (Iϕ) frequency with respect to the output frequency of PLL circuit. If a prohibited value is specified, subsequent operation is not guaranteed.<br>000: ×1<br>001: ×1/2<br>010: ×1/3<br>011: ×1/4<br>100: ×1/8<br>Other than above: Setting prohibited |
| 11 to 9 | BFC[2:0] | 011 | R/W | Bus Clock (Bϕ) Frequency Division Ratio<br>Specify the division ratio of the bus clock (Bϕ) frequency with respect to the output frequency of PLL circuit. If a prohibited value is specified, subsequent operation is not guaranteed.<br>000: ×1<br>001: ×1/2<br>010: ×1/3<br>011: ×1/4<br>100: ×1/8<br>Other than above: Setting prohibited |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 8 to 6 | PFC[2:0] | 011 | R/W | Peripheral Clock (Pϕ) Frequency Division Ratio<br>Specify the division ratio of the peripheral clock (Pϕ) frequency with respect to the output frequency of PLL circuit. If a prohibited value is specified, subsequent operation is not guaranteed.<br>000: ×1<br>001: ×1/2<br>010: ×1/3<br>011: ×1/4<br>100: ×1/8<br>Other than above: Setting prohibited |
| 5 to 3 | MIFC[2:0] | 011 | R/W | MTU2S Clock (MIϕ) Frequency Division Ratio<br>Specify the division ratio of the MTU2S clock (MIϕ) frequency with respect to the output frequency of PLL circuit. If a prohibited value is specified, subsequent operation is not guaranteed.<br>000: ×1<br>001: ×1/2<br>010: ×1/3<br>011: ×1/4<br>100: ×1/8<br>Other than above: Setting prohibited |
| 2 to 0 | MPFC[2:0] | 011 | R/W | MTU2 Clock (MPϕ) Frequency Division Ratio<br>Specify the division ratio of the MTU2 clock (MPϕ) frequency with respect to the output frequency of PLL circuit. If a prohibited value is specified, subsequent operation is not guaranteed.<br>000: ×1<br>001: ×1/2<br>010: ×1/3<br>011: ×1/4<br>100: ×1/8<br>Other than above: Setting prohibited |

### 4.4.2 Oscillation Stop Detection Control Register (OSCCR)

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

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

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

## 4.5 Changing Frequency

Selecting division ratios for the frequency divider can change the frequencies of the internal clock (Iϕ), bus clock (Bϕ), peripheral clock (Pϕ), MTU2S clock (MIϕ), and MTU2 clock (MPϕ). This is controlled by software through the frequency control register (FRQCR). The following describes how to specify the frequencies.

1. In the initial state, IFC2 to IFC0 = H'011 (×1/4), BFC2 to BFC0 = H'011 (×1/4), PFC2 to PFC0 = H'011 (×1/4), MIFC2 to MIFC0 = H'011 (×1/4), and MPFC2 to MPFC0 = H'011 (×1/4).
2. Stop all modules except the CPU, on-chip ROM, and on-chip RAM.
3. Set the desired values in bits IFC2 to IFC0, BFC2 to BFC0, PFC2 to PFC0, MIFC2 to MIFC0, and MPFC2 to MPFC0 bits. Since the frequency multiplication ratio in the PLL circuit is fixed at ×8, the frequencies are determined only be selecting division ratios. When specifying the frequencies, satisfy the following condition: internal clock (Iϕ) ≥ bus clock (Bϕ) ≥ peripheral clock (Pϕ). When using the MTU2S clock and MTU2 clock, specify the frequencies to satisfy the following condition: internal clock (Iϕ) ≥ MTU2S clock (MIϕ) ≥ MTU2 clock (MPϕ) ≥ peripheral clock (Pϕ) and bus clock (Bϕ) ≥ MTU2 clock (MPϕ). Code to rewrite values of FRQCR should be executed in the on-chip ROM or on-chip RAM.
4. After an instruction to rewrite FRQCR has been issued, the actual clock frequencies will change after (1 to 24n) cyc + 11Bϕ + 7Pϕ.

   n: Division ratio specified by the BFC bit in FRQCR (1, 1/2, 1/3, 1/4, or 1/8)

   cyc: Clock obtained by dividing EXTAL by 8 with the PLL.

Note: (1 to 24n) depends on the internal state.

## 4.6 Oscillator

Clock pulses can be supplied from a connected crystal resonator or an external clock.

### 4.6.1 Connecting Crystal Resonator

A crystal resonator can be connected as shown in figure 4.2. Use the damping resistance (Rd) listed in table 4.6. Use a crystal resonator that has a resonance frequency of 5 to 12.5 MHz. It is recommended to consult the crystal resonator manufacturer concerning the compatibility of the crystal resonator and the LSI.

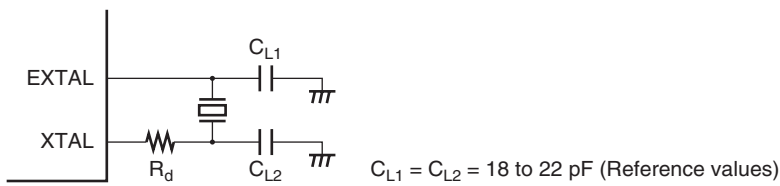



**Figure 4.2 Connection of Crystal Resonator (Example)**

**Table 4.6 Damping Resistance Values (Reference Values)**

| Frequency (MHz) | 5 | 8 | 10 | 12.5 |
|---|---|---|---|---|
| Rd (Ω) (Reference values) | 500 | 200 | 0 | 0 |

Figure 4.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator with the characteristics listed in table 4.7.

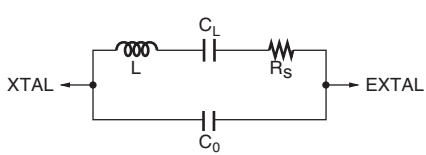



**Figure 4.3 Crystal Resonator Equivalent Circuit**

**Table 4.7 Crystal Resonator Characteristics**

| Frequency (MHz) | 5 | 8 | 10 | 12.5 |
|---|---|---|---|---|
| Rs Max. (Ω) (Reference values) | 120 | 80 | 60 | 50 |
| $C_0$ Max. (pF) (Reference values) | 7 | 7 | 7 | 7 |

### 4.6.2 External Clock Input Method

Figure 4.4 shows an example of an external clock input connection. In this case, make the external clock high level to stop it when in software standby mode. During operation, make the external input clock frequency 5 to 12.5 MHz.

When leaving the XTAL pin open, make sure the parasitic capacitance is less than 10 pF.

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

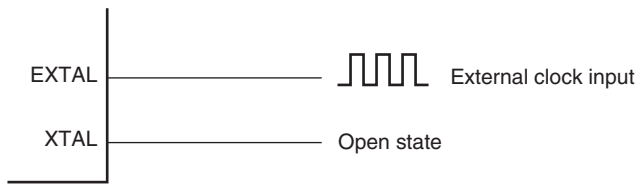



**Figure 4.4 Example of External Clock Connection**

## 4.7 Function for Detecting Oscillator Stop

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

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

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

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

## 4.8 Usage Notes

### 4.8.1 Note on Crystal Resonator

A sufficient evaluation at the user's site is necessary to use the LSI, by referring the resonator connection examples shown in this section, because various characteristics related to the crystal resonator are closely linked to the user's board design. As the oscillator circuit's circuit constant will depend on the resonator and the floating capacitance of the mounting circuit, the value of each external circuit's component 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 oscillator pin.

### 4.8.2 Notes on Board Design

Measures against radiation noise are taken in this LSI. If further reduction in radiation noise is needed, it is recommended to use a multiple layer board and provide a layer exclusive to the system ground.

When using a crystal resonator, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins. Do not route any signal lines near the oscillator circuitry as shown in figure 4.5. Otherwise, correct oscillation can be interfered by induction.

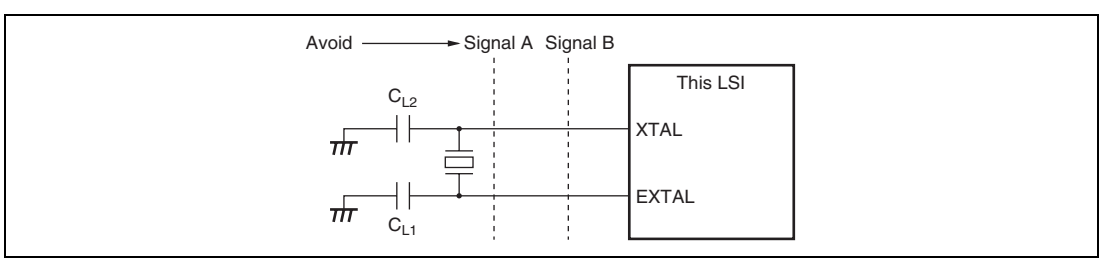
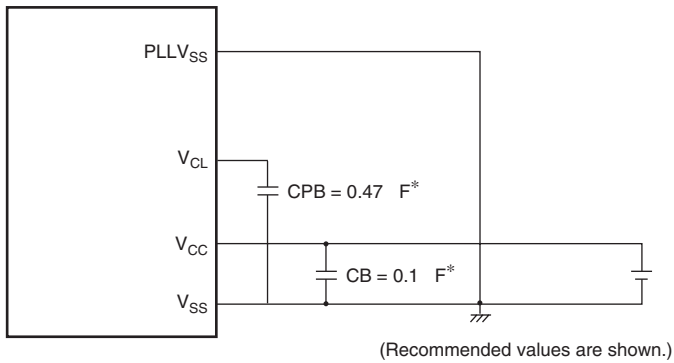



**Figure 4.5 Cautions for Oscillator Circuit Board Design**

A circuitry shown in figure 4.6 is recommended as an external circuitry around the PLL. Separate the PLL power lines (PLLVss) and the system power lines (Vcc, Vss) at the board power supply source, and be sure to insert bypass capacitors CB and CPB close to the pins.




Note: * CB and CPB are laminated ceramic type.

**Figure 4.6 Recommended External Circuitry around PLL**

# Section 5 Exception Handling

## 5.1 Overview

### 5.1.1 Types of Exception Handling and Priority

Exception handling is started by four sources: resets, address errors, interrupts and instructions and have the priority, as shown in table 5.1. When several exceptions are detected at once, they are processed according to the priority.

**Table 5.1 Types of Exceptions and Priority**

| Exception | Exception Source | Priority |
|---|---|---|
| Reset | Power-on reset | High |
| | Manual reset | ↑ |
| Interrupt | User break (break before instruction execution)*3 | |
| Address error | CPU address error (instruction fetch) | |
| Instruction | General illegal instructions (undefined code) | |
| | Illegal slot instruction (undefined code placed immediately after a delayed branch instruction*1 or instruction that changes the PC value*2) | |
| | Trap instruction (TRAPA instruction) | |
| Address error | CPU address error (data access) | |
| Interrupt | User break (break after instruction execution or operand break)*3 | |
| Address error | DTC address error (data access) | |
| Interrupt | NMI | |
| | IRQ | ↓ |
| | On-chip peripheral modules | Low |

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF.
2. Instructions that change the PC value: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, LDC Rm,SR, LDC.L @Rm+,SR.
3. SH7136 and SH7137 only.

### 5.1.2 Exception Handling Operations

The exceptions are detected and the exception handling starts according to the timing shown in table 5.2.

**Table 5.2 Timing for Exception Detection and Start of Exception Handling**

| Exception | | Timing of Source Detection and Start of Exception Handling |
|---|---|---|
| Reset | Power-on reset | Started when the $\overline{\text{RES}}$ pin changes from low to high or when the WDT overflows. |
| | Manual reset | Started when the $\overline{\text{MRES}}$ pin changes from low to high or when the WDT overflows. |
| Address error | | Detected during the instruction decode stage and started after the execution of the current instruction is completed. |
| Interrupt | | |
| Instruction | Trap instruction | Started by the execution of the TRAPA instruction. |
| | General illegal instructions | Started when an undefined code placed at other than a delay slot (immediately after a delayed branch instruction) is decoded. |
| | Illegal slot instructions | Started when an undefined code placed at a delay slot (immediately after a delayed branch instruction) or an instruction that changes the PC value is detected. |

When exception handling starts, the CPU operates

**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 from the address H'00000000 and SP from the address H'00000004 when a power-on reset. PC from the address H'00000008 and SP from the address H'0000000C when a manual reset.). For details, see section 5.1.3, Exception Handling Vector Table. H'00000000 is then written to the vector base register (VBR), and H'F (B'1111) is written to the interrupt mask bits (I3 to I0) in the status register (SR). The program starts from the PC address fetched from the exception handling vector table.

**Exception Handling Triggered by Address Error, Interrupt, and Instruction:** SR and PC are saved to the stack indicated by R15. For interrupt exception handling, the interrupt priority level is written to the interrupt mask bits (I3 to I0) in SR. For address error and instruction exception handling, bits I3 to I0 are not affected. The start address is then fetched from the exception handling vector table and the program starts from that address.

### 5.1.3 Exception Handling Vector Table

Before exception handling starts, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception handling 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. The vector table addresses are calculated from these vector numbers and vector table address offsets. During exception handling, the start addresses of the exception handling routines are fetched from the exception handling vector table that 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 Vector Numbers and Vector Table Address Offsets**

| Exception Handling Source | | Vector Number | 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 for system use) | | 5 | H'00000014 to H'00000017 |
| Illegal slot instruction | | 6 | H'00000018 to H'0000001B |
| (Reserved for system use) | | 7 | H'0000001C to H'0000001F |
| | | 8 | H'00000020 to H'00000023 |
| CPU address error | | 9 | H'00000024 to H'00000027 |
| DTC address error | | 10 | H'00000028 to H'0000002B |
| Interrupt | NMI | 11 | H'0000002C to H'0000002F |
| | User break*[1] | 12 | H'00000030 to H'00000033 |
| (Reserved for system use) | | 13 | H'00000034 to H'00000037 |
| | | : | : |
| | | 31 | H'0000007C to H'0000007F |
| Trap instruction (user vector) | | 32 | H'00000080 to H'00000083 |
| | | : | : |
| | | 63 | H'000000FC to H'000000FF |

| Exception Handling Source | | Vector Number | Vector Table Address Offset |
|---|---|---|---|
| Interrupt | IRQ0 | 64 | H'00000100 to H'00000103 |
| | IRQ1 | 65 | H'00000104 to H'00000107 |
| | IRQ2 | 66 | H'00000108 to H'0000010B |
| | IRQ3 | 67 | H'0000010C to H'0000010F |
| (Reserved for system use) | | 68 | H'00000110 to H'00000113 |
| | | 69 | H'00000114 to H'00000117 |
| | | 70 | H'00000118 to H'0000011B |
| | | 71 | H'0000011C to H'0000011F |
| On-chip peripheral module*2 | | 72 | H'00000120 to H'00000123 |
| | | : | : |
| | | 255 | H'000003FC to H'000003FF |

Notes: 1. SH7136 and SH7137 only.
2. For details on the vector numbers and vector table address offsets of on-chip peripheral module interrupts, see table 6.3 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)<br>= (vector number) × 4 |
| Address errors, interrupts, instructions | Vector table address = VBR + (vector table address offset)<br>= VBR + (vector number) × 4 |

Notes: 1. VBR: Vector base register
2. Vector table address offset: See table 5.3.
3. Vector number: See table 5.3.

## 5.2 Resets

### 5.2.1 Types of Resets

Resets have priority over any exception source. There are two types of resets: power-on resets and manual resets. As table 5.5 shows, both types of resets initialize the internal status of the CPU. In power-on resets, all registers of the on-chip peripheral modules are initialized; in manual resets, they are not.

**Table 5.5 Reset Status**

| | Conditions for Transition to Reset State | | | Internal State | | |
|---|---|---|---|---|---|---|
| **Type** | **$\overline{RES}$** | **WDT Overflow** | **$\overline{MRES}$** | **CPU, INTC** | **On-Chip Peripheral Module** | **POE, PFC, I/O Port** |
| Power-on reset | Low | — | — | Initialized | Initialized | Initialized |
| | High | Overflow | High | Initialized | Initialized | Initialized |
| Manual reset | High | Not overflowed | Low | Initialized | Not initialized | Not initialized |

### 5.2.2 Power-On Reset

**Power-On Reset by $\overline{RES}$ Pin:** When the $\overline{RES}$ pin is driven low, this LSI enters the power-on reset state. To reliably reset this LSI, the $\overline{RES}$ pin should be kept low for at least the oscillation settling time when applying the power or when in standby mode (when the clock is halted) or at least 20 tcyc when the clock is operating. During the power-on reset state, CPU internal states and all registers of on-chip peripheral modules are initialized. See appendix A, Pin States, for the status of individual pins during power-on reset mode.

In the power-on reset state, power-on reset exception handling starts when driving the $\overline{RES}$ pin high after driving the pin low for the given time. 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 and the interrupt mask bits (I3 to I0) of the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in PC and SP, then the program starts.

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

**Power-On Reset by WDT:** When WTCNT of the WDT overflows while a setting is made so that a power-on reset can be generated in watchdog timer mode of the WDT, this LSI enters the power-on reset state.

The frequency control register (FRQCR) in the clock pulse generator (CPG) and the watchdog timer (WDT) registers are not initialized by the reset signal generated by the WDT (these registers are only initialized by a power-on reset from the $\overline{\text{RES}}$ pin).

If a reset caused by the signal input on the $\overline{\text{RES}}$ pin and a reset caused by a WDT overflow occur simultaneously, the $\overline{\text{RES}}$ pin reset has priority, and the WOVF bit in WTCSR is cleared to 0. When the power-on reset exception handling caused by the WDT is started, 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 and the interrupt mask bits (I3 to I0) of the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in the PC and SP, then the program starts.

### 5.2.3 Manual Reset

When the $\overline{\text{RES}}$ pin is high and the $\overline{\text{MRES}}$ pin is driven low, the LSI becomes to be a manual reset state. To reliably reset the LSI, the $\overline{\text{MRES}}$ pin should be kept at low for at least the duration of the oscillation settling time that is set in WDT when in software standby mode (when the clock is halted) or at least 20 $t_{cyc}$ when the clock is operating. During manual reset, the CPU internal status is initialized. Registers of on-chip peripheral modules are not initialized. When the LSI enters manual reset status in the middle of a bus cycle, manual reset exception processing does not start until the bus cycle has ended. Thus, manual resets do not abort bus cycles. However, once $\overline{\text{MRES}}$ is driven low, hold the low level until the CPU becomes to be a manual reset mode after the bus cycle ends. (Keep at low level for at least the longest bus cycle). See appendix A, Pin States, for the status of individual pins during manual reset mode.

In the manual reset status, manual reset exception processing starts when the $\overline{\text{MRES}}$ pin is first kept low for a set period of time and then returned to high. The CPU will then operate in the same procedures as described for power-on resets.

## 5.3 Address Errors

### 5.3.1 Address Error Sources

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

**Table 5.6 Bus Cycles and Address Errors**

| Bus Cycle | | | |
|---|---|---|---|
| **Type** | **Bus Master** | **Bus Cycle Description** | **Address Errors** |
| Instruction fetch | CPU | Instruction fetched from even address | None (normal) |
| | | Instruction fetched from odd address | Address error occurs |
| | | Instruction fetched from a space other than on-chip peripheral module space | None (normal) |
| | | Instruction fetched from on-chip peripheral module space | Address error occurs |
| | | Instruction fetched from external memory space in single chip mode | Address error occurs |
| Data read/write | CPU or DTC | Word data accessed from even address | None (normal) |
| | | Word data accessed from odd address | Address error occurs |
| | | Longword data accessed from a longword boundary | None (normal) |
| | | Longword data accessed from other than a long-word boundary | Address error occurs |
| | | Byte or word data accessed in on-chip peripheral module space | None (normal) |
| | | Longword data accessed in 16-bit on-chip peripheral module space | None (normal) |
| | | Longword data accessed in 8-bit on-chip peripheral module space | None (normal) |
| | | External memory space accessed when in single chip mode | Address error occurs |

### 5.3.2 Address Error Exception Source

When an address error exception is generated, the bus cycle which caused the address error ends, the current instruction finishes, and then the address error exception handling starts. The CPU operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value to be saved is the start address of the instruction which caused an address error exception. When the instruction that caused the exception is placed in the delay slot, the address of the delayed branch instruction which is placed immediately before the delay slot.
3. The start address of the exception handling routine is fetched from the exception handling vector table that corresponds to the generated address error, and the program starts executing from that address. This branch is not a delayed branch.

## 5.4 Interrupts

### 5.4.1 Interrupt Sources

Table 5.7 shows the sources that start the interrupt exception handling. They are NMI, user break, IRQ, and on-chip peripheral modules.

**Table 5.7 Interrupt Sources**

| Type | Request Source | Number of Sources |
|---|---|---|
| NMI | NMI pin (external input) | 1 |
| User break* | User break controller (UBC) | 1 |
| IRQ | IRQ0 to IRQ3 pins (external input) | 4 |
| On-chip peripheral module | Multi-function timer pulse unit 2 (MTU2) | 28 |
| | Multi-function timer pulse unit 2S (MTU2S) | 13 |
| | Data transfer controller (DTC) | 1 |
| | Watchdog timer (WDT) | 1 |
| | A/D converter (A/D_0 and A/D_1) | 2 |
| | Compare match timer (CMT_0 and CMT_1) | 2 |
| | Serial communication interface (SCI_0, SCI_1, and SCI_2) | 12 |
| | Port output enable (POE) | 3 |
| | Synchronous serial communication unit (SSU) | 3 |
| | $I^2C$ bus interface 2 ($I^2C2$) | 5 |
| | Controller area network (RCAN-ET) | 5 |

Note: * SH7136 and SH7137 only.

All interrupt sources are given different vector numbers and vector table address offsets. For details on vector numbers and vector table address offsets, see table 6.3 in section 6, Interrupt Controller (INTC).

### 5.4.2 Interrupt Priority

The interrupt priority is predetermined. When multiple interrupts occur simultaneously (overlapped interruptions), the interrupt controller (INTC) determines their relative priorities and starts the exception handling according to the results.

The priority 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 priority level of the user break interrupt is 15. IRQ interrupt and on-chip peripheral module interrupt priority levels can be set freely using the interrupt priority registers A, D to F, and H to M (IPRA, IPRD to IPRF, and IPRH to IPRM) of the INTC as shown in table 5.8. The priority levels that can be set are 0 to 15. Level 16 cannot be set. For details on IPRA, IPRD to IPRF, and IPRH to IPRM, see section 6.3.4, Interrupt Priority Registers A, D to F, and H to M (IPRA, IPRD to IPRF, and IPRH to IPRM).

**Table 5.8 Interrupt Priority**

| Type | Priority Level | Comment |
|---|---|---|
| NMI | 16 | Fixed priority level. Cannot be masked. |
| User break* | 15 | Fixed priority level. Can be masked. |
| IRQ | 0 to 15 | Set with interrupt priority registers A, D to F, and H to M (IPRA, IPRD to IPRF, and IPRH to IPRM). |
| On-chip peripheral module | 0 to 15 | |

Note: * SH7136 and SH7137 only.

### 5.4.3 Interrupt Exception Handling

When an interrupt occurs, the interrupt controller (INTC) ascertains its priority level. 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 bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, exception handling begins. In interrupt exception handling, the CPU saves SR and the program counter (PC) to the stack. The priority level of the accepted interrupt is written to bits I3 to I0 in SR. Although the priority level of the NMI is 16, the value set in bits I3 to I0 is H'F (level 15). Next, the start address of the exception handling routine is fetched from the exception handling vector table for the accepted interrupt, and program execution branches to that address and the program starts. For details on the interrupt exception handling, see section 6.6, Interrupt Operation.

## 5.5 Exceptions Triggered by Instructions

### 5.5.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by the trap instruction, illegal slot instructions, and general illegal instructions, as shown in table 5.9.

**Table 5.9 Types of Exceptions Triggered by Instructions**

| Type | Source Instruction | Comment |
|---|---|---|
| Trap instruction | TRAPA | — |
| Illegal slot instructions* | Undefined code placed immediately after a delayed branch instruction (delay slot) or instructions that changes the PC value | Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF<br>Instructions that changes the PC value: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, LDC Rm,SR, LDC.L @Rm+,SR |
| General illegal instructions* | Undefined code anywhere besides in a delay slot | — |

Note: * The operation is not guaranteed when undefined instructions other than H'F000 to H'FFFF are decoded.

### 5.5.2 Trap Instructions

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

1. The status register (SR) is saved to the stack.
2. 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.
3. The CPU reads the start address of the exception handling routine from the exception handling vector table that corresponds to the vector number specified in the TRAPA instruction, program execution branches to that address, and then the program starts. This branch is not a delayed branch.

### 5.5.3 Illegal Slot Instructions

An instruction placed immediately after a delayed branch instruction is called "instruction placed in a delay slot". When the instruction placed in the delay slot is an undefined code, illegal slot exception handling starts after the undefined code is decoded. Illegal slot exception handling also starts when an instruction that changes the program counter (PC) value is placed in a delay slot and the instruction is decoded. The CPU handles an illegal slot instruction as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the target address of the delayed branch instruction immediately before the undefined code or the instruction that rewrites the PC.
3. The start address of the exception handling routine is fetched from the exception handling vector table that corresponds to the exception that occurred. Program execution branches to that address and the program starts. This branch is not a delayed branch.

### 5.5.4 General Illegal Instructions

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

## 5.6 Cases when Exceptions are Accepted

When an exception other than resets occurs during decoding the instruction placed in a delay slot or immediately after an interrupt disabled instruction, it may not be accepted and be held shown in table 5.10. In this case, when an instruction which accepts an interrupt request is decoded, the exception is accepted.

**Table 5.10 Delay Slot Instructions, Interrupt Disabled Instructions, and Exceptions**

| | Exception | | | | |
|---|---|---|---|---|---|
| **Occurrence Timing** | **Address Error** | **General Illegal Instruction** | **Slot Illegal Instruction** | **Trap Instruction** | **Interrupt** |
| Instruction in delay slot | ×*2 | — | ×*2 | — | ×*3 |
| Immediately after interrupt disabled instruction*1 | √ | √ | √ | √ | ×*4 |

[Legend]

√: Accepted

×: Not accepted

—: Does not occur

Notes: 1. Interrupt disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, and STS.L

2. An exception is accepted before the execution of a delayed branch instruction. However, when an address error or a slot illegal instruction exception occurs in the delay slot of the RTE instruction, correct operation is not guaranteed.

3. An exception is accepted after a delayed branch (between instructions in the delay slot and the branch destination).

4. An exception is accepted after the execution of the next instruction of an interrupt disabled instruction (before the execution two instructions after an interrupt disabled instruction).

## 5.7 Stack States after Exception Handling Ends

The stack states after exception handling ends are shown in table 5.11.

**Table 5.11 Stack Status after Exception Handling Ends**

| Types | Stack State |
|---|---|
| Address error (when the instruction that caused an exception is placed in the delay slot) | SP → Address of delayed branch instruction (32 bits); SR (32 bits) |
| Address error (other than above) | SP → Address of instruction that caused exception (32 bits); SR (32 bits) |
| Interrupt | SP → Address of instruction after executed instruction (32 bits); SR (32 bits) |
| Trap instruction | SP → Address of instruction after TRAPA instruction (32 bits); SR (32 bits) |

| Types | Stack State |
|---|---|
| Illegal slot instruction | SP → Address of delayed branch instruction (32 bits); SR (32 bits) |
| General illegal instruction | SP → Address of general illegal instruction (32 bits); SR (32 bits) |

## 5.8 Usage Notes

### 5.8.1 Value of Stack Pointer (SP)

The SP value must always be a multiple of 4. If it is not, an address error will occur when the stack is accessed during exception handling.

### 5.8.2 Value of Vector Base Register (VBR)

The VBR value must always be a multiple of 4. If it is not, an address error will occur when the stack is accessed during exception handling.

### 5.8.3 Address Errors Caused by Stacking for Address Error Exception Handling

When the SP value is not a multiple of 4, an address error will occur when stacking for exception handling (interrupts, etc.) and address error exception handling will start after the first exception handling is ended. Address errors will 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 passed to the handling routine for address error exception and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle (write) is executed. When stacking the SR and PC values, the SP values for both are subtracted by 4, therefore, the SP value is still not a multiple of 4 after the stacking. The address value output during stacking is the SP value whose lower two bits are cleared to 0. So the write data stacked is undefined.

### 5.8.4 Notes on Slot Illegal Instruction Exception Handling

Some specifications on slot illegal instruction exception handling in this LSI differ from those of the conventional SH-2.

- Conventional SH-2: Instructions LDC Rm,SR and LDC.L @Rm+,SR are not subject to the slot illegal instructions.
- This LSI: Instructions LDC Rm,SR and LDC.L @Rm+,SR are subject to the slot illegal instructions.

The supporting status on our software products regarding this note is as follows:

**Compiler**

This instruction is not allocated in the delay slot in the compiler V.4 and its subsequent versions.

**Real-time OS for μITRON specifications**

1. HI7000/4, HI-SH7
   This instruction does not exist in the delay slot within the OS.
2. HI7000
   This instruction is in part allocated to the delay slot within the OS, which may cause the slot illegal instruction exception handling in this LSI.
3. Others
   The slot illegal instruction exception handling may be generated in this LSI in a case where the instruction is described in assembler or when the middleware of the object is introduced.

Note that a check-up program (checker) to pick up this instruction is available on our website. Download and utilize this checker as needed.

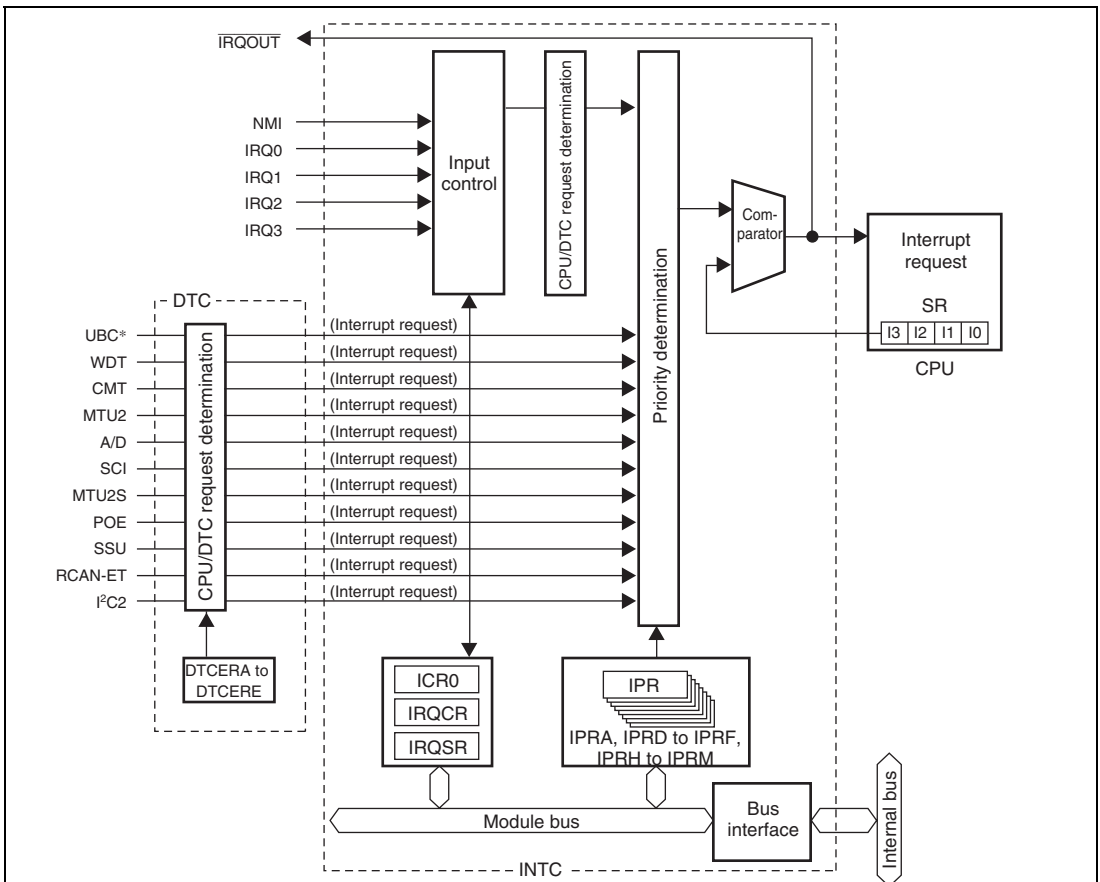
# Section 6 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU.

## 6.1 Features

- 16 levels of interrupt priority
- NMI noise canceller function
- Occurrence of interrupt can be reported externally ($\overline{\text{IRQOUT}}$ pin)

Figure 6.1 shows a block diagram of the INTC.




[Legend]

UBC: User break controller
WDT: Watchdog timer
CMT: Compare match timer
SCI: Serial communication interface
MTU2: Multi-function timer pulse unit 2
MTU2S: Multi-function timer pulse unit 2S
A/D: A/D converter
POE: Port output enable
DTC: Data transfer controller
SSU: Synchronous serial communication unit
RCAN-ET: Controller area network
$I^2C2$: $I^2C$ bus interface 2

ICR0: Interrupt control register 0
IRQCR: IRQ control register
IRQSR: IRQ status register
IPRA, IPRD to IPRF, IPRH to IPRM: Interrupt priority registers A, D to F, and H to M
SR: Status register
DTCERA to DTCERE: DTC enable registers A to E

Note: * SH7136 and SH7137 only.

**Figure 6.1 Block Diagram of INTC**

## 6.2 Input/Output Pins

Table 6.1 shows the INTC pin configuration.

**Table 6.1 Pin Configuration**

| Name | Symbol | I/O | Function |
|---|---|---|---|
| Non-maskable interrupt input pin | NMI | Input | Input of non-maskable interrupt request signal |
| Interrupt request input pins | IRQ0 to IRQ3 | Input | Input of maskable interrupt request signals |
| Interrupt request output pin | $\overline{\text{IRQOUT}}$ | Output | Output of notification signal when an interrupt has occurred |

## 6.3 Register Descriptions

The interrupt controller has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 25, List of Registers.

**Table 6.2 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Interrupt control register 0 | ICR0 | R/W | H'x000 | H'FFFFE900 | 8, 16 |
| IRQ control register | IRQCR | R/W | H'0000 | H'FFFFE902 | 8, 16 |
| IRQ status register | IRQSR | R/W | H'Fx00 | H'FFFFE904 | 8, 16 |
| Interrupt priority register A | IPRA | R/W | H'0000 | H'FFFFE906 | 8, 16 |
| Interrupt priority register D | IPRD | R/W | H'0000 | H'FFFFE982 | 16 |
| Interrupt priority register E | IPRE | R/W | H'0000 | H'FFFFE984 | 16 |
| Interrupt priority register F | IPRF | R/W | H'0000 | H'FFFFE986 | 16 |
| Interrupt priority register H | IPRH | R/W | H'0000 | H'FFFFE98A | 16 |
| Interrupt priority register I | IPRI | R/W | H'0000 | H'FFFFE98C | 16 |
| Interrupt priority register J | IPRJ | R/W | H'0000 | H'FFFFE98E | 16 |
| Interrupt priority register K | IPRK | R/W | H'0000 | H'FFFFE990 | 16 |
| Interrupt priority register L | IPRL | R/W | H'0000 | H'FFFFE992 | 16 |
| Interrupt priority register M | IPRM | R/W | H'0000 | H'FFFFE994 | 16 |

### 6.3.1 Interrupt Control Register 0 (ICR0)

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

| 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: * The initial value is 1 when the level on the NMI pin is high, and 0 when the level on the pin is low.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | NMIL | * | R | NMI Input Level<br>Indicates the state of the signal input to the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified.<br>0: State of the NMI input is low<br>1: State of the NMI input is high |
| 14 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 8 | NMIE | 0 | R/W | NMI Edge Select<br>0: Interrupt request is detected on the falling edge of the NMI input<br>1: Interrupt request is detected on the rising edge of the NMI input |
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

### 6.3.2 IRQ Control Register (IRQCR)

IRQCR is a 16-bit register that sets the input signal detection mode of the external interrupt input pins IRQ0 to IRQ3.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | IRQ31S | IRQ30S | IRQ21S | IRQ20S | IRQ11S | IRQ10S | IRQ01S | IRQ00S |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | 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<br>These bits are always read as 0. The write value should always be 0. |
| 7 | IRQ31S | 0 | R/W | IRQ3 Sense Select |
| 6 | IRQ30S | 0 | R/W | Set the interrupt request detection mode for pin IRQ3.<br>00: Interrupt request is detected at the low level of pin IRQ3<br>01: Interrupt request is detected at the falling edge of pin IRQ3<br>10: Interrupt request is detected at the rising edge of pin IRQ3<br>11: Interrupt request is detected at both the falling and rising edges of pin IRQ3 |
| 5 | IRQ21S | 0 | R/W | IRQ2 Sense Select |
| 4 | IRQ20S | 0 | R/W | Set the interrupt request detection mode for pin IRQ2.<br>00: Interrupt request is detected at the low level of pin IRQ2<br>01: Interrupt request is detected at the falling edge of pin IRQ2<br>10: Interrupt request is detected at the rising edge of pin IRQ2<br>11: Interrupt request is detected at both the falling and rising edges of pin IRQ2 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | IRQ11S | 0 | R/W | IRQ1 Sense Select |
| 2 | IRQ10S | 0 | R/W | Set the interrupt request detection mode for pin IRQ1.<br>00: Interrupt request is detected at the low level of pin IRQ1<br>01: Interrupt request is detected at the falling edge of pin IRQ1<br>10: Interrupt request is detected at the rising edge of pin IRQ1<br>11: Interrupt request is detected at both the falling and rising edges of pin IRQ1 |
| 1 | IRQ01S | 0 | R/W | IRQ0 Sense Select |
| 0 | IRQ00S | 0 | R/W | Set the interrupt request detection mode for pin IRQ0.<br>00: Interrupt request is detected at the low level of pin IRQ0<br>01: Interrupt request is detected at the falling edge of pin IRQ0<br>10: Interrupt request is detected at the rising edge of pin IRQ0<br>11: Interrupt request is detected at both the falling and rising edges of pin IRQ0 |

### 6.3.3 IRQ Status register (IRQSR)

IRQSR is a 16-bit register that indicates the states of the external interrupt input pins IRQ0 to IRQ3 and the status of interrupt request.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | IRQ3L | IRQ2L | IRQ1L | IRQ0L | - | - | - | - | IRQ3F | IRQ2F | IRQ1F | IRQ0F |
| Initial value: | 1 | 1 | 1 | 1 | * | * | * | * | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |

Note: * The initial value is 1 when the level on the corresponding IRQ pin is high, and 0 when the level on the pin is low.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 12 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. |
| 11 | IRQ3L | * | R | Indicates the state of pin IRQ3.<br>0: State of pin IRQ3 is low<br>1: State of pin IRQ3 is high |
| 10 | IRQ2L | * | R | Indicates the state of pin IRQ2.<br>0: State of pin IRQ2 is low<br>1: State of pin IRQ2 is high |
| 9 | IRQ1L | * | R | Indicates the state of pin IRQ1.<br>0: State of pin IRQ1 is low<br>1: State of pin IRQ1 is high |
| 8 | IRQ0L | * | R | Indicates the state of pin IRQ0.<br>0: State of pin IRQ0 is low<br>1: State of pin IRQ0 is high |
| 7 to 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | IRQ3F | 0 | R/W | Indicates the status of an IRQ3 interrupt request.<br>• When level detection mode is selected<br>0: An IRQ3 interrupt has not been detected<br>[Clearing condition]<br>Driving pin IRQ3 high<br>1: An IRQ3 interrupt has been detected<br>[Setting condition]<br>Driving pin IRQ3 low<br>• When edge detection mode is selected<br>0: An IRQ3 interrupt has not been detected<br>[Clearing conditions]<br>— Writing 0 after reading IRQ3F = 1<br>— Accepting an IRQ3 interrupt<br>1: An IRQ3 interrupt request has been detected<br>[Setting condition]<br>Detecting the specified edge of pin IRQ3 |
| 2 | IRQ2F | 0 | R/W | Indicates the status of an IRQ2 interrupt request.<br>• When level detection mode is selected<br>0: An IRQ2 interrupt has not been detected<br>[Clearing condition]<br>Driving pin IRQ2 high<br>1: An IRQ2 interrupt has been detected<br>[Setting condition]<br>Driving pin IRQ2 low<br>• When edge detection mode is selected<br>0: An IRQ2 interrupt has not been detected<br>[Clearing conditions]<br>— Writing 0 after reading IRQ2F = 1<br>— Accepting an IRQ2 interrupt<br>1: An IRQ2 interrupt request has been detected<br>[Setting condition]<br>Detecting the specified edge of pin IRQ2 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | IRQ1F | 0 | R/W | Indicates the status of an IRQ1 interrupt request.<br>• When level detection mode is selected<br>0: An IRQ1 interrupt has not been detected<br>[Clearing condition]<br>Driving pin IRQ1 high<br>1: An IRQ1 interrupt has been detected<br>[Setting condition]<br>Driving pin IRQ1 low<br>• When edge detection mode is selected<br>0: An IRQ1 interrupt has not been detected<br>[Clearing conditions]<br>— Writing 0 after reading IRQ1F = 1<br>— Accepting an IRQ1 interrupt<br>1: An IRQ1 interrupt request has been detected<br>[Setting condition]<br>Detecting the specified edge of pin IRQ1 |
| 0 | IRQ0F | 0 | R/W | Indicates the status of an IRQ0 interrupt request.<br>• When level detection mode is selected<br>0: An IRQ0 interrupt has not been detected<br>[Clearing condition]<br>Driving pin IRQ0 high<br>1: An IRQ0 interrupt has been detected<br>[Setting condition]<br>Driving pin IRQ0 low<br>• When edge detection mode is selected<br>0: An IRQ0 interrupt has not been detected<br>[Clearing conditions]<br>— Writing 0 after reading IRQ0F = 1<br>— Accepting an IRQ0 interrupt<br>1: An IRQ0 interrupt request has been detected<br>[Setting condition]<br>Detecting the specified edge of pin IRQ0 |

Note: * The initial value is 1 when the level on the corresponding IRQ pin is high, and 0 when the level on the pin is low.

### 6.3.4 Interrupt Priority Registers A, D to F, and H to M (IPRA, IPRD to IPRF, and IPRH to IPRM)

Interrupt priority registers are ten 16-bit readable/writable registers that set priority levels from 0 to 15 for interrupts except NMI. For the correspondence between interrupt request sources and IPR, refer to table 6.3. Each of the corresponding interrupt priority ranks are established by setting a value from H'0 to H'F in each of the four-bit groups 15 to 12, 11 to 8, 7 to 4 and 3 to 0. Reserved bits that are not assigned should be set H'0 (B'0000).

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IPR[15:12] | | | | IPR[11:8] | | | | IPR[7:4] | | | | IPR[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 |
|---|---|---|---|---|
| 15 to 12 | IPR[15:12] | 0000 | R/W | Set priority levels for the corresponding interrupt source.<br>0000: Priority level 0 (lowest)<br>0001: Priority level 1<br>0010: Priority level 2<br>0011: Priority level 3<br>0100: Priority level 4<br>0101: Priority level 5<br>0110: Priority level 6<br>0111: Priority level 7<br>1000: Priority level 8<br>1001: Priority level 9<br>1010: Priority level 10<br>1011: Priority level 11<br>1100: Priority level 12<br>1101: Priority level 13<br>1110: Priority level 14<br>1111: Priority level 15 (highest) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 11 to 8 | IPR[11:8] | 0000 | R/W | Set priority levels for the corresponding interrupt source.<br>0000: Priority level 0 (lowest)<br>0001: Priority level 1<br>0010: Priority level 2<br>0011: Priority level 3<br>0100: Priority level 4<br>0101: Priority level 5<br>0110: Priority level 6<br>0111: Priority level 7<br>1000: Priority level 8<br>1001: Priority level 9<br>1010: Priority level 10<br>1011: Priority level 11<br>1100: Priority level 12<br>1101: Priority level 13<br>1110: Priority level 14<br>1111: Priority level 15 (highest) |
| 7 to 4 | IPR[7:4] | 0000 | R/W | Set priority levels for the corresponding interrupt source.<br>0000: Priority level 0 (lowest)<br>0001: Priority level 1<br>0010: Priority level 2<br>0011: Priority level 3<br>0100: Priority level 4<br>0101: Priority level 5<br>0110: Priority level 6<br>0111: Priority level 7<br>1000: Priority level 8<br>1001: Priority level 9<br>1010: Priority level 10<br>1011: Priority level 11<br>1100: Priority level 12<br>1101: Priority level 13<br>1110: Priority level 14<br>1111: Priority level 15 (highest) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 to 0 | IPR[3:0] | 0000 | R/W | Set priority levels for the corresponding interrupt source.<br>0000: Priority level 0 (lowest)<br>0001: Priority level 1<br>0010: Priority level 2<br>0011: Priority level 3<br>0100: Priority level 4<br>0101: Priority level 5<br>0110: Priority level 6<br>0111: Priority level 7<br>1000: Priority level 8<br>1001: Priority level 9<br>1010: Priority level 10<br>1011: Priority level 11<br>1100: Priority level 12<br>1101: Priority level 13<br>1110: Priority level 14<br>1111: Priority level 15 (highest) |

Note: Name in the tables above is represented by a general name. Name in the list of register is, on the other hand, represented by a module name.

## 6.4 Interrupt Sources

### 6.4.1 External Interrupts

There are four types of interrupt sources: User break, NMI, IRQ, and on-chip peripheral modules. Individual interrupts are given priority levels (0 to 16, with 0 the lowest and 16 the highest). Giving an interrupt a priority level of 0 masks it.

**NMI Interrupt:** The NMI interrupt is given a priority level of 16 and is always accepted. An NMI interrupt is detected at the edge of the pins. Use the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) to select either the rising or falling edge. In the NMI interrupt exception handler, the interrupt mask level bits (I3 to I0) in the status register (SR) are set to level 15.

**IRQ3 to IRQ0 Interrupts:** IRQ interrupts are requested by input from pins IRQ0 to IRQ3. Use the IRQ sense select bits (IRQ31S, IRQ30S to IRQ01S, and IRQ00S) in the IRQ control register (IRQCR) to select the detection mode from low level detection, falling edge detection, rising edge detection, and both edge detection for each pin. The priority level can be set from 0 to 15 for each pin using the interrupt priority register A (IPRA).

In the case that the low level detection is selected, an interrupt request signal is sent to the INTC while the IRQ pin is driven low. The interrupt request signal stops to be sent to the INTC when the IRQ pin becomes high. It is possible to confirm that an interrupt is requested by reading the IRQ flags (IRQ3F to IRQ0F) in the IRQ status register (IRQSR).

In the case that the edge detection is selected, an interrupt request signal is sent to the INTC when the following change on the IRQ pin is detected: from high to low in falling edge detection mode, from low to high in rising edge detection mode, and from low to high or from high to low in both edge detection mode. The IRQ interrupt request by detecting the change on the pin is held until the interrupt request is accepted. It is possible to confirm that an IRQ interrupt request has been detected by reading the IRQ flags (IRQ3F to IRQ0F) in the IRQ status register (IRQSR). An IRQ interrupt request by detecting the change on the pin can be withdrawn by writing 0 to an IRQ flag after reading 1.

In the IRQ interrupt exception handling, the interrupt mask bits (I3 to I0) in the status register (SR) are set to the priority level value of the accepted IRQ interrupt. Figure 6.2 shows the block diagram of the IRQ3 to IRQ0 interrupts.

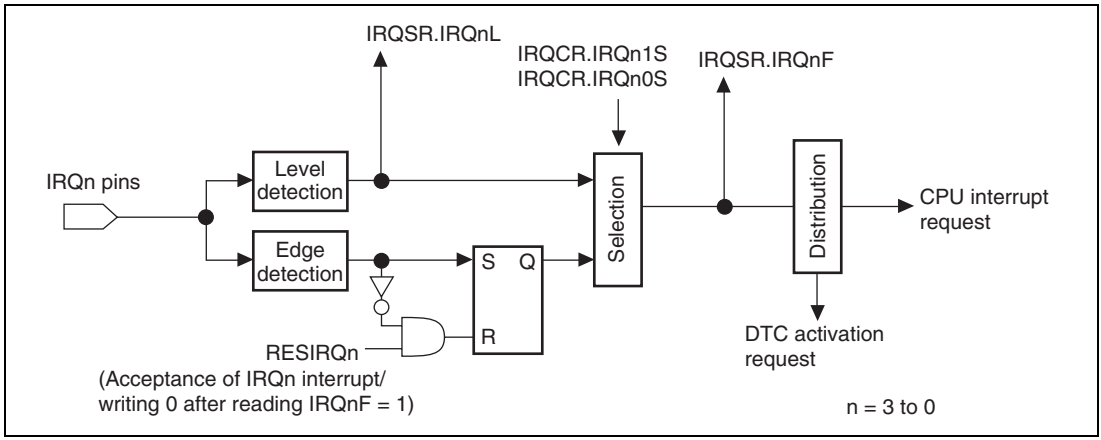



**Figure 6.2 Block Diagram of IRQ3 to IRQ0 Interrupts Control**

### 6.4.2 On-Chip Peripheral Module Interrupts

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

Since a different interrupt vector is allocated to each interrupt source, the exception handling routine does not have to decide which interrupt has occurred. Priority levels between 0 and 15 can be allocated to individual on-chip peripheral modules in interrupt priority registers D to F and H to M (IPRD to IPRF and IPRH to IPRM). On-chip peripheral module interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

### 6.4.3 User Break Interrupt (SH7136 and SH7137 only)

A user break interrupt has a priority level of 15, and occurs when the break condition set in the user break controller (UBC) is satisfied. User break interrupt requests are detected by edge and are held until accepted. User break interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to level 15. For more details on the user break interrupt, see section 7, User Break Controller (UBC).

## 6.5 Interrupt Exception Handling Vector Table

Table 6.3 lists interrupt sources, their vector numbers, vector table address offsets, and interrupt priorities.

Individual interrupt sources are allocated to different vector numbers and vector table address offsets. Vector table addresses are calculated from the vector numbers and vector table address offsets. For interrupt exception handling, the start address of the exception handling routine is fetched from the vector table address in the vector table. For the details on calculation of vector table addresses, see table 5.4 in section 5, Exception Handling.

IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers A, D to F and H to M (IPRA, IPRD to IPRF, and IPRH to IPRM). However, when interrupt sources whose priority levels are allocated with the same IPR are requested, the interrupt of the smaller vector number has priority. This priority cannot be changed. Priority levels of IRQ interrupts and on-chip peripheral module interrupts are initialized to level 0 at a power-on reset. If the same priority level is allocated to two or more interrupt sources and interrupts from those sources occur simultaneously, they are processed by the default priority order shown in table 6.3.

**Table 6.3 Interrupt Exception Handling Vectors and Priorities**

| Interrupt Source | Name | Vector No. | Vector Table Starting Address | IPR | Default Priority |
|---|---|---|---|---|---|
| User break*[1] | | 12 | H'00000030 | — | High |
| External pin | NMI | 11 | H'0000002C | — | ↑ |
| | IRQ0 | 64 | H'00000100 | IPRA15 to IPRA12 | |
| | IRQ1 | 65 | H'00000104 | IPRA11 to IPRA8 | |
| | IRQ2 | 66 | H'00000108 | IPRA7 to IPRA4 | |
| | IRQ3 | 67 | H'0000010C | IPRA3 to IPRA0 | |
| MTU2_0 | TGIA_0 | 88 | H'00000160 | IPRD15 to IPRD12 | |
| | TGIB_0 | 89 | H'00000164 | | |
| | TGIC_0 | 90 | H'00000168 | | |
| | TGID_0 | 91 | H'0000016C | | |
| | TCIV_0 | 92 | H'00000170 | IPRD11 to IPRD8 | |
| | TGIE_0 | 93 | H'00000174 | | |
| | TGIF_0 | 94 | H'00000178 | | Low |

| Interrupt Source | Name | Vector No. | Vector Table Starting Address | IPR | Default Priority |
|---|---|---|---|---|---|
| MTU2_1 | TGIA_1 | 96 | H'00000180 | IPRD7 to IPRD4 | High |
| | TGIB_1 | 97 | H'00000184 | | |
| | TCIV_1 | 100 | H'00000190 | IPRD3 to IPRD0 | |
| | TCIU_1 | 101 | H'00000194 | | |
| MTU2_2 | TGIA_2 | 104 | H'000001A0 | IPRE15 to IPRE12 | |
| | TGIB_2 | 105 | H'000001A4 | | |
| | TCIV_2 | 108 | H'000001B0 | IPRE11 to IPRE8 | |
| | TCIU_2 | 109 | H'000001B4 | | |
| MTU2_3 | TGIA_3 | 112 | H'000001C0 | IPRE7 to IPRE4 | |
| | TGIB_3 | 113 | H'000001C4 | | |
| | TGIC_3 | 114 | H'000001C8 | | |
| | TGID_3 | 115 | H'000001CC | | |
| | TCIV_3 | 116 | H'000001D0 | IPRE3 to IPRE0 | |
| MTU2_4 | TGIA_4 | 120 | H'000001E0 | IPRF15 to IPRF12 | |
| | TGIB_4 | 121 | H'000001E4 | | |
| | TGIC_4 | 122 | H'000001E8 | | |
| | TGID_4 | 123 | H'000001EC | | |
| | TCIV_4 | 124 | H'000001F0 | IPRF11 to IPRF8 | |
| MTU2_5 | TGIU_5 | 128 | H'00000200 | IPRF7 to IPRF4 | |
| | TGIV_5 | 129 | H'00000204 | | |
| | TGIW_5 | 130 | H'00000208 | | |
| POE (MTU2) | OEI1 | 132 | H'00000210 | IPRF3 to IPRF0 | |
| | OEI3 | 133 | H'00000214 | | |
| $I^2C2$*2 | IINAKI | 156 | H'00000270 | IPRH11 to IPRH8 | |
| MTU2S_3 | TGIA_3S | 160 | H'00000280 | IPRH7 to IPRH4 | |
| | TGIB_3S | 161 | H'00000284 | | |
| | TGIC_3S | 162 | H'00000288 | | |
| | TGID_3S | 163 | H'0000028C | | |
| | TCIV_3S | 164 | H'00000290 | IPRH3 to IPRH0 | Low |

| Interrupt Source | Name | Vector No. | Vector Table Starting Address | IPR | Default Priority |
|---|---|---|---|---|---|
| MTU2S_4 | TGIA_4S | 168 | H'000002A0 | IPRI15 to IPRI12 | High |
| | TGIB_4S | 169 | H'000002A4 | | |
| | TGIC_4S | 170 | H'000002A8 | | |
| | TGID_4S | 171 | H'000002AC | | |
| | TCIV_4S | 172 | H'000002B0 | IPRI11 to IPRI8 | |
| MTU2S_5 | TGIU_5S | 176 | H'000002C0 | IPRI7 to IPRI4 | |
| | TGIV_5S | 177 | H'000002C4 | | |
| | TGIW_5S | 178 | H'000002C8 | | |
| POE (MTU2S) | OEI2 | 180 | H'000002D0 | IPRI3 to IPRI0 | |
| CMT_0 | CMI_0 | 184 | H'000002E0 | IPRJ15 to IPRJ12 | |
| CMT_1 | CMI_1 | 188 | H'000002F0 | IPRJ11 to IPRJ8 | |
| WDT | ITI | 196 | H'00000310 | IPRJ3 to IPRJ0 | |
| A/D_0 | ADI_3 | 208 | H'00000340 | IPRK7 to IPRK4 | |
| A/D_1 | ADI_4 | 212 | H'00000350 | IPRK3 to IPRK0 | |
| SCI_0 | ERI_0 | 216 | H'00000360 | IPRL15 to IPRL12 | |
| | RXI_0 | 217 | H'00000364 | | |
| | TXI_0 | 218 | H'00000368 | | |
| | TEI_0 | 219 | H'0000036C | | |
| SCI_1 | ERI_1 | 220 | H'00000370 | IPRL11 to IPRL8 | |
| | RXI_1 | 221 | H'00000374 | | |
| | TXI_1 | 222 | H'00000378 | | |
| | TEI_1 | 223 | H'0000037C | | |
| SCI_2 | ERI_2 | 224 | H'00000380 | IPRL7 to IPRL4 | |
| | RXI_2 | 225 | H'00000384 | | |
| | TXI_2 | 226 | H'00000388 | | |
| | TEI_2 | 227 | H'0000038C | | |
| SSU | SSERI | 232 | H'000003A0 | IPRM15 to IPRM12 | |
| | SSRXI | 233 | H'000003A4 | | |
| | SSTXI | 234 | H'000003A8 | | Low |

| Interrupt Source | Name | Vector No. | Vector Table Starting Address | IPR | Default Priority |
|---|---|---|---|---|---|
| $I^2C$*2 | IITEI | 236 | H'000003B0 | IPRM11 to IPRM8 | High |
| | IISTPI | 237 | H'000003B4 | | ↑ |
| | IITXI | 238 | H'000003B8 | | |
| | IIRXI | 239 | H'000003BC | | |
| RCAN-ET_0 | ERS_0 | 240 | H'000003C0 | IPRM7 to IPRM4 | |
| | OVR_0 | 241 | H'000003C4 | | |
| | RM0_0 | 242 | H'000003C8 | | |
| | RM1_0 | | | | |
| | SLE_0 | 243 | H'000003CC | | Low |

Notes: 1. SH7136 and SH7137 only.

2. Of the $I^2C2$ interrupts, the vector address for the IINAKI interrupt is separated from others.

## 6.6 Interrupt Operation

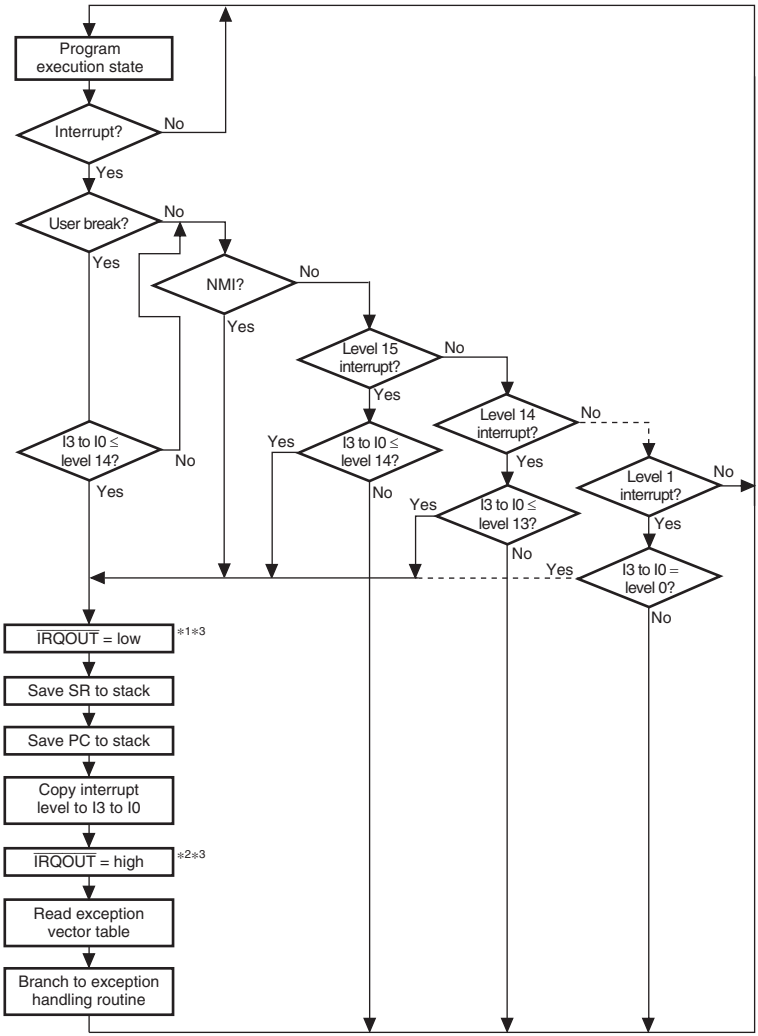
### 6.6.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 6.3 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest priority interrupt from interrupt requests sent, according to the priority levels set in interrupt priority registers A, D to F, and H to M (IPRA, IPRD to IPRF, and IPRH to IPRM). Interrupts that have lower-priority than that of the selected interrupt are ignored*. If interrupts that have the same priority level or interrupts within a same module occur simultaneously, the interrupt with the highest priority is selected according to the default priority shown in table 6.3.
3. The interrupt controller compares the priority level of the selected interrupt request with the interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the priority level of the selected request is equal to or less than the level set in bits I3 to I0, the request is ignored. If the priority level of the selected request is higher than the level in bits I3 to I0, the interrupt controller accepts the request and sends an interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low level is output from the $\overline{\text{IRQOUT}}$ pin.
5. The CPU detects the interrupt request sent from the interrupt controller in the decode stage of an instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception handling.
6. SR and PC are saved onto the stack.
7. The priority level of the accepted interrupt is copied to bits (I3 to I0) in SR.
8. When the accepted interrupt is sensed by level or is from an on-chip peripheral module, a high level is output from the $\overline{\text{IRQOUT}}$ pin. When the accepted interrupt is sensed by edge, a high level is output from the $\overline{\text{IRQOUT}}$ pin at the moment when the CPU starts interrupt exception processing instead of instruction execution as noted in 5. above. However, if the interrupt controller accepts an interrupt with a higher priority than the interrupt just to be accepted, the $\overline{\text{IRQOUT}}$ pin holds low level.
9. The CPU reads the start address of the exception handling routine from the exception vector table for the accepted interrupt, branches to that address, and starts executing the program. This branch is not a delayed branch.

Notes: The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt source that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, confirm that it has been cleared, and then execute an RTE instruction.

* Interrupt requests that are designated as edge-detect type are held pending until the interrupt requests are accepted. IRQ interrupts, however, can be cancelled by accessing the IRQ status register (IRQSR). Interrupts held pending due to edge detection are cleared by a power-on reset or a manual reset.



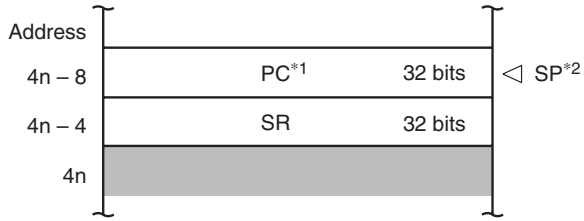

Notes: I3 to I0 are interrupt mask bits in the status register (SR) of the CPU

1. $\overline{\text{IRQOUT}}$ is the same signal as the interrupt request signal to the CPU (see figure 6.1). Therefore, $\overline{\text{IRQOUT}}$ is output when the request priority level is higher than the level in bits I3–I0 of SR.
2. When the accepted interrupt is sensed by edge, a high level is output from the $\overline{\text{IRQOUT}}$ pin at the moment when the CPU starts interrupt exception processing instead of instruction execution (namely, before saving SR to stack). However, if the interrupt controller accepts an interrupt with a higher priority than the interrupt just to be accepted and has output an interrupt request to the CPU, the $\overline{\text{IRQOUT}}$ pin holds low level.
3. The $\overline{\text{IRQOUT}}$ pin change timing depends on a frequency dividing ratio between the internal (Iϕ) and bus (Bϕ) clocks. This flowchart shows that the frequency dividing ratios of the internal (Iϕ) and bus (Bϕ) clocks are the same.

**Figure 6.3 Interrupt Sequence Flowchart**

### 6.6.2 Stack after Interrupt Exception Handling

Figure 6.4 shows the stack after interrupt exception handling.




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

**Figure 6.4 Stack after Interrupt Exception Handling**

## 6.7 Interrupt Response Time

Table 6.4 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 of the interrupt handling routine begins.

## Table 6.4 Interrupt Response Time

| Item | | Number of Cycles: NMI | IRQ | Peripheral Modules | Remarks |
|---|---|---|---|---|---|
| DTC active judgment | | — | 2 × Bcyc | 1 × Pcyc | |
| Interrupt priority decision and comparison with mask bits in SR | | 1 × Icyc + 2 × Pcyc | 1 × Icyc + 1 × Pcyc | 1 × Icyc + 2 × Pcyc | |
| Wait for completion of sequence currently being executed by CPU | | X (≥ 0) | X (≥ 0) | X (≥ 0) | The longest sequence is for interrupt or address-error exception handling (X = 7 × Icyc + m1 + m2 + m3 + m4). If an interrupt-masking instruction follows, however, the time may be even longer. |
| Time from start of interrupt exception handling until fetch of first instruction of exception handling routine starts | | 8 × Icyc + m1 + m2 + m3 | 8 × Icyc + m1 + m2 + m3 | 8 × Icyc + m1 + m2 + m3 | Performs the saving PC and SR, and vector address fetch. |
| Interrupt response time | Total: | 9 × Icyc + 2 × Pcyc + m1 + m2 + m3 + X | 9 × Icyc + 1 × Pcyc +2 × Bcyc + m1 + m2 + m3 + X | 9 × Icyc + 3 × Pcyc + m1 + m2 + m3 + X | |
| | Minimum*: | 12 × Icyc + 2 × Pcyc | 12 × Icyc + 1 × Pcyc + 2 × Bcyc | 12 × Icyc + 3 × Pcyc | SR, PC, and vector table are all in on-chip RAM. |
| | Maximum: | 16 × Icyc + 2 × Pcyc + 2 × (m1 + m2 + m3) + m4 | 16 × Icyc + 1 × Pcyc + 2 × Bcyc + 2 × (m1 + m2 + m3) + m4 | 16 × Icyc + 3 × Pcyc + 2 × (m1 + m2 + m3) + m4 | |

Notes: * In the case that m1 = m2 = m3 = m4 = 1 × Icyc.
m1 to m4 are the number of cycles needed for the following memory accesses.
m1: SR save (longword write)
m2: PC save (longword write)
m3: Vector address read (longword read)
m4: Fetch first instruction of interrupt service routine

## 6.8 Data Transfer with Interrupt Request Signals

The following data transfers can be done using interrupt request signals:

- Activate DTC only; CPU interrupts depend on DTC settings

The INTC masks a CPU interrupt when the corresponding DTCE bit is 1. The conditions for clearing DTCE and interrupt source flag are shown below.

DTCE clear condition = DTC transfer end • DTCECLR
Interrupt source flag clear condition = DTC transfer end • $\overline{\text{DTCECLR}}$
where DTCECLR = DISEL + counter 0

Figures 6.5 and 6.6 show control block diagrams.

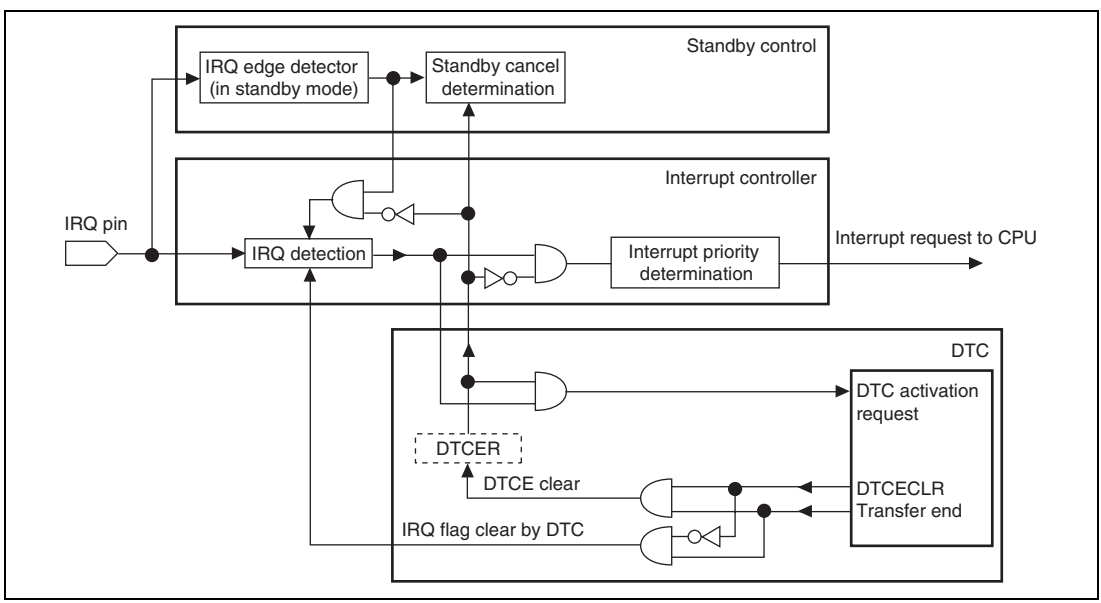



**Figure 6.5 IRQ Interrupt Control Block Diagram**

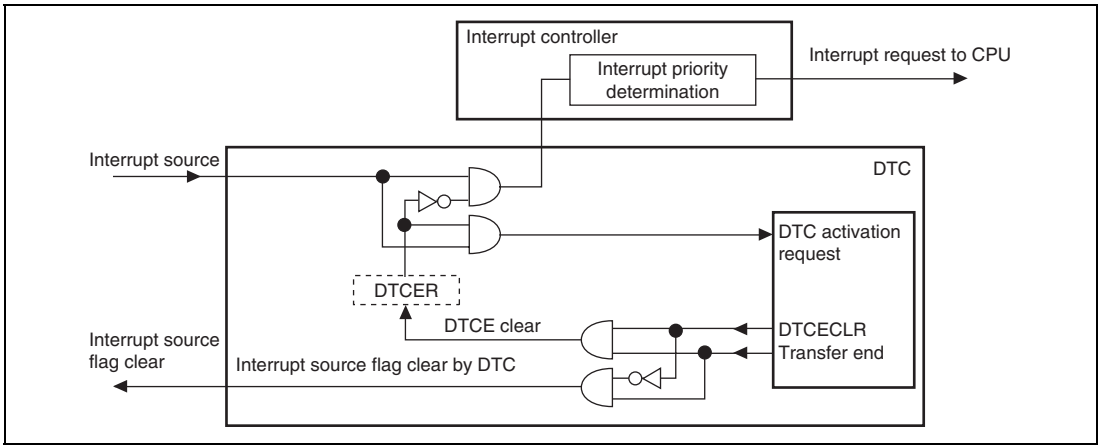



**Figure 6.6 On-Chip Module Interrupt Control Block Diagram**

### 6.8.1 Handling Interrupt Request Signals as Sources for DTC Activation and CPU Interrupts

1. For DTC, set the corresponding DTCE bits and DISEL bits to 1.
2. When an interrupt occurs, an activation request is sent to the DTC.
3. When completing a data transfer, the DTC clears the DTCE bit to 0 and sends an interrupt request to the CPU. The activation source is not cleared.
4. The CPU clears the interrupt source in the interrupt handling routine then checks the transfer counter value. When the transfer counter value is not 0, the CPU sets the DTCE bit to 1 and allows the next data transfer. If the transfer counter value = 0, the CPU performs the necessary end processing in the interrupt processing routine.

### 6.8.2 Handling Interrupt Request Signals as Sources for DTC Activation, but Not CPU Interrupts

1. For DTC, set the corresponding DTCE bits to 1 and clear the DISEL bits to 0.
2. When an interrupt occurs, an activation request is sent to the DTC.
3. When completing a data transfer, the DTC clears the activation source. No interrupt request is sent to the CPU because the DTCE bit is held at 1.
4. However, when the transfer counter value = 0, the DTCE bit is cleared to 0 and an interrupt request is sent to the CPU.
5. The CPU performs the necessary end processing in the interrupt handling routine.

### 6.8.3 Handling Interrupt Request Signals as Sources for CPU Interrupts, but Not DTC Activation

1. For DTC, clear the corresponding DTCE bits to 0.
2. When an interrupt occurs, an interrupt request is sent to the CPU.
3. The CPU clears the interrupt source and performs the necessary processing in the interrupt handling routine.

## 6.9 Usage Note

The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt source that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, confirm that it has been cleared, and then execute an RTE instruction.

# Section 7 User Break Controller (UBC) (SH7136 and SH7137 only)

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. Break conditions that can be set in the UBC are instruction fetch or data read/write access, data size, data contents, address value, and stop timing in the case of instruction fetch.

## 7.1 Features

The UBC has the following features:

1. The following break comparison conditions can be set.

   Number of break channels: two channels (channels A and B)

   User break can be requested as either the independent or sequential condition on channels A and B (sequential break setting: channel A and then channel B match with break conditions, but not in the same bus cycle).

   — Address

     Comparison bits are maskable in 1-bit units.

     One of the two address buses (L-bus address (LAB) and I-bus address (IAB)) can be selected.

   — Data

     32-bit maskable.

     One of the two data buses (L-bus data (LDB) and I-bus data (IDB)) can be selected.

   — Bus cycle

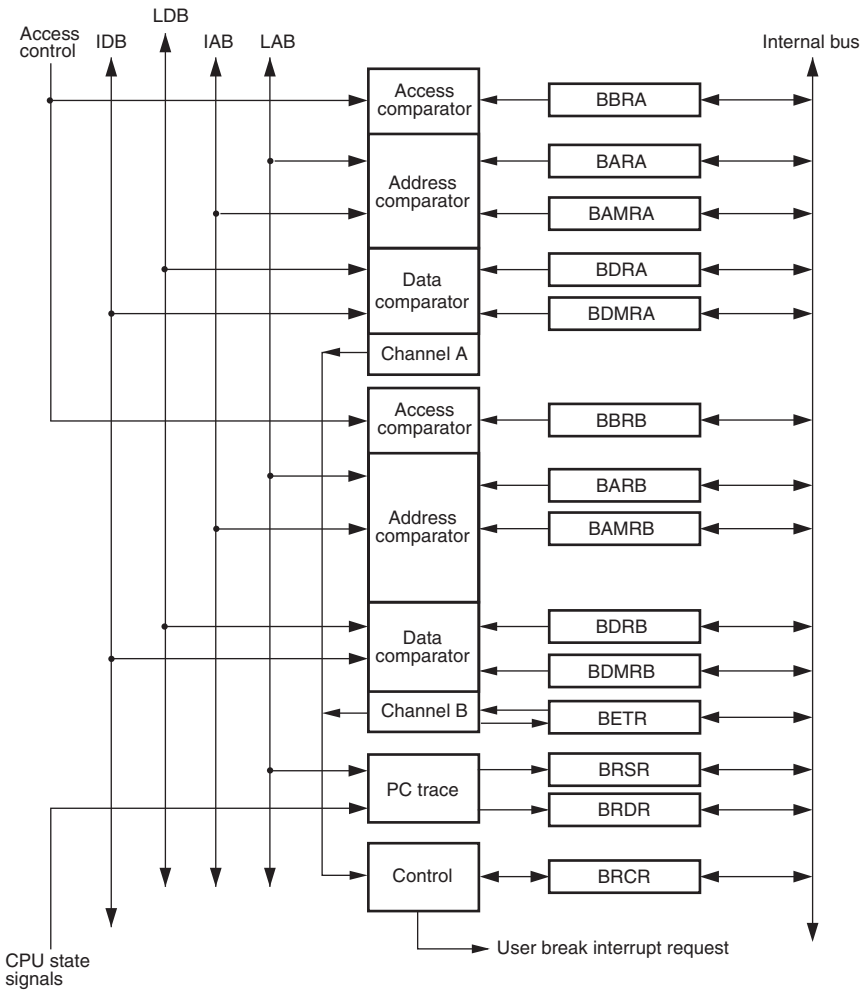
     Instruction fetch or data access

   — Read/write

   — Operand size

     Byte, word, and longword
2. A user-designed user-break interrupt exception processing routine can be run.
3. In an instruction fetch cycle, whether a user break is set before or after execution of an instruction can be selected.
4. Maximum repeat times for the break condition (only for channel B): $2^{12} - 1$ times.
5. Four pairs of branch source/destination buffers.

Figure 7.1 shows a block diagram of the UBC.




[Legend]

| | | | |
|---|---|---|---|
| BBRA: | Break bus cycle register A | BDRB: | Break data register B |
| BARA: | Break address register A | BDMRB: | Break data mask register B |
| BAMRA: | Break address mask register A | BETR: | Execution times break register |
| BDRA: | Break data register A | BRSR: | Branch source register |
| BDMRA: | Break data mask register A | BRDR: | Branch destination register |
| BBRB: | Break bus cycle register B | BRCR: | Break control register |
| BARB: | Break address register B | | |
| BAMRB: | Break address mask register B | | |

**Figure 7.1 Block Diagram of UBC**

## 7.2 Input/Output Pins

Table 7.1 shows the UBC pin configuration.

**Table 7.1 Pin Configuration**

| Pin Name | Symbol | I/O | Function |
|---|---|---|---|
| User break trigger output | $\overline{\text{UBCTRG}}$ | Output | UBC condition match trigger output pin. |

## 7.3 Register Descriptions

The user break controller has the following registers. For details on register addresses and register states during each processing, refer to section 25, List of Registers.

**Table 7.2 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Break address register A | BARA | R/W | H'00000000 | H'FFFFF300 | 32 |
| Break address mask register A | BAMRA | R/W | H'00000000 | H'FFFFF304 | 32 |
| Break bus cycle register A | BBRA | R/W | H'0000 | H'FFFFF308 | 16 |
| Break data register A | BDRA | R/W | H'00000000 | H'FFFFF310 | 32 |
| Break data mask register A | BDMRA | R/W | H'00000000 | H'FFFFF314 | 32 |
| Break address register B | BARB | R/W | H'00000000 | H'FFFFF320 | 32 |
| Break address mask register B | BAMRB | R/W | H'00000000 | H'FFFFF324 | 32 |
| Break bus cycle register B | BBRB | R/W | H'0000 | H'FFFFF328 | 16 |
| Break data register B | BDRB | R/W | H'00000000 | H'FFFFF330 | 32 |
| Break data mask register B | BDMRB | R/W | H'00000000 | H'FFFFF334 | 32 |
| Break control register | BRCR | R/W | H'00000000 | H'FFFFF3C0 | 32 |
| Branch source register | BRSR | R | H'0xxxxxxx | H'FFFFF3D0 | 32 |
| Branch destination register | BRDR | R | H'0xxxxxxx | H'FFFFF3D4 | 32 |
| Execution times break register | BETR | R/W | H'0000 | H'FFFFF3DC | 16 |

### 7.3.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register. BARA specifies the address used as a break condition in channel A.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAA31 | BAA30 | BAA29 | BAA28 | BAA27 | BAA26 | BAA25 | BAA24 | BAA23 | BAA22 | BAA21 | BAA20 | BAA19 | BAA18 | BAA17 | BAA16 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAA15 | BAA14 | BAA13 | BAA12 | BAA11 | BAA10 | BAA9 | BAA8 | BAA7 | BAA6 | BAA5 | BAA4 | BAA3 | BAA2 | BAA1 | BAA0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | BAA31 to BAA 0 | All 0 | R/W | Break Address A<br>Store the address on the LAB or IAB specifying break conditions of channel A. |

### 7.3.2 Break Address Mask Register A (BAMRA)

BAMRA is a 32-bit readable/writable register. BAMRA specifies bits masked in the break address specified by BARA.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAMA31 | BAMA30 | BAMA29 | BAMA28 | BAMA27 | BAMA26 | BAMA25 | BAMA24 | BAMA23 | BAMA22 | BAMA21 | BAMA20 | BAMA19 | BAMA18 | BAMA17 | BAMA16 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAMA15 | BAMA14 | BAMA13 | BAMA12 | BAMA11 | BAMA10 | BAMA9 | BAMA8 | BAMA7 | BAMA6 | BAMA5 | BAMA4 | BAMA3 | BAMA2 | BAMA1 | BAMA0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | BAMA31 to BAMA 0 | All 0 | R/W | Break Address Mask A<br>Specify bits masked in the channel A break address bits specified by BARA (BAA31 to BAA0).<br>0: Break address bit BAAn of channel A is included in the break condition<br>1: Break address bit BAAn of channel A is masked and is not included in the break condition<br>Note: n = 31 to 0 |

### 7.3.3 Break Bus Cycle Register A (BBRA)

BBRA is a 16-bit readable/writable register, which specifies (1) bus master for I bus cycle, (2) L bus cycle or I bus cycle, (3) instruction fetch or data access, (4) read or write, and (5) operand size in the break conditions of channel A.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | CPA[2:0] | | | CDA[1:0] | | IDA[1:0] | | RWA[1:0] | | SZA[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/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 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 10 to 8 | CPA[2:0] | 000 | R/W | Bus Master Select A for I Bus<br>Select the bus master when the I bus is selected as the bus cycle of the channel A break condition. However, when the L bus is selected as the bus cycle, the setting of the CPA2 to CPA0 bits is disabled.<br>000: Condition comparison is not performed<br>xx1: The CPU cycle is included in the break condition<br>x1x: Setting prohibited<br>1xx: The DTC cycle is included in the break condition |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | CDA[1:0] | 00 | R/W | L Bus Cycle/I Bus Cycle Select A<br>Select the L bus cycle or I bus cycle as the bus cycle of the channel A break condition.<br>00: Condition comparison is not performed<br>01: The break condition is the L bus cycle<br>10: The break condition is the I bus cycle<br>11: The break condition is the L bus cycle |
| 5, 4 | IDA[1:0] | 00 | R/W | Instruction Fetch/Data Access Select A<br>Select the instruction fetch cycle or data access cycle as the bus cycle of the channel A break condition.<br>00: Condition comparison is not performed<br>01: The break condition is the instruction fetch cycle<br>10: The break condition is the data access cycle<br>11: The break condition is the instruction fetch cycle or data access cycle |
| 3, 2 | RWA[1:0] | 00 | R/W | Read/Write Select A<br>Select the read cycle or write cycle as the bus cycle of the channel A break condition.<br>00: Condition comparison is not performed<br>01: The break condition is the read cycle<br>10: The break condition is the write cycle<br>11: The break condition is the read cycle or write cycle |
| 1, 0 | SZA[1:0] | 00 | R/W | Operand Size Select A<br>Select the operand size of the bus cycle for the channel A break condition.<br>00: The break condition does not include operand size<br>01: The break condition is byte access<br>10: The break condition is word access<br>11: The break condition is longword access<br>Note: When specifying the operand size, specify the size which matches the address boundary. |

[Legend]

x: Don't care.

### 7.3.4 Break Data Register A (BDRA)

BDRA is a 32-bit readable/writable register. The control bits CDA1 and CDA0 in BBRA select one of two data buses for break condition A.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BDA31 | BDA30 | BDA29 | BDA28 | BDA27 | BDA26 | BDA25 | BDA24 | BDA23 | BDA22 | BDA21 | BDA20 | BDA19 | BDA18 | BDA17 | BDA16 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BDA15 | BDA14 | BDA13 | BDA12 | BDA11 | BDA10 | BDA9 | BDA8 | BDA7 | BDA6 | BDA5 | BDA4 | BDA3 | BDA2 | BDA1 | BDA0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | BDA31 to BDA0 | All 0 | R/W | Break Data Bit A<br>Stores data which specifies a break condition in channel A.<br>If the I bus is selected in BBRA, the break data on IDB is set in BDA31 to BDA0.<br>If the L bus is selected in BBRA, the break data on LDB is set in BDA31 to BDA0. |

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.

2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDRA as the break data.

### 7.3.5 Break Data Mask Register A (BDMRA)

BDMRA is a 32-bit readable/writable register. BDMRA specifies bits masked in the break data specified by BDRA.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BDMA31 | BDMA30 | BDMA29 | BDMA28 | BDMA27 | BDMA26 | BDMA25 | BDMA24 | BDMA23 | BDMA22 | BDMA21 | BDMA20 | BDMA19 | BDMA18 | BDMA17 | BDMA16 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BDMA15 | BDMA14 | BDMA13 | BDMA12 | BDMA11 | BDMA10 | BDMA9 | BDMA8 | BDMA7 | BDMA6 | BDMA5 | BDMA4 | BDMA3 | BDMA2 | BDMA1 | BDMA0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | BDMA31 to BDMA 0 | All 0 | R/W | Break Data Mask A<br>Specifies bits masked in the break data of channel A specified by BDRA (BDA31 to BDA0).<br>0: Break data BDAn of channel A is included in the break condition<br>1: Break data BDAn of channel A is masked and is not included in the break condition<br>Note: n = 31 to 0 |

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.

2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDMRA as the break mask data in BDRA.

### 7.3.6 Break Address Register B (BARB)

BARB is a 32-bit readable/writable register. BARB specifies the address used as a break condition in channel B. Control bits CDB1 and CDB0 in BBRB select one of the two address buses for break condition B.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAB31 | BAB30 | BAB29 | BAB28 | BAB27 | BAB26 | BAB25 | BAB24 | BAB23 | BAB22 | BAB21 | BAB20 | BAB19 | BAB18 | BAB17 | BAB16 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAB15 | BAB14 | BAB13 | BAB12 | BAB11 | BAB10 | BAB9 | BAB8 | BAB7 | BAB6 | BAB5 | BAB4 | BAB3 | BAB2 | BAB1 | BAB0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | BAB31 to BAB 0 | All 0 | R/W | Break Address B<br>Stores an address which specifies a break condition in channel B.<br>If the I bus or L bus is selected in BBRB, an IAB or LAB address is set in BAB31 to BAB0. |

### 7.3.7 Break Address Mask Register B (BAMRB)

BAMRB is a 32-bit readable/writable register. BAMRB specifies bits masked in the break address specified by BARB.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAMB31 | BAMB30 | BAMB29 | BAMB28 | BAMB27 | BAMB26 | BAMB25 | BAMB24 | BAMB23 | BAMB22 | BAMB21 | BAMB20 | BAMB19 | BAMB18 | BAMB17 | BAMB16 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAMB15 | BAMB14 | BAMB13 | BAMB12 | BAMB11 | BAMB10 | BAMB9 | BAMB8 | BAMB7 | BAMB6 | BAMB5 | BAMB4 | BAMB3 | BAMB2 | BAMB1 | BAMB0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | BAMB31 to BAMB 0 | All 0 | R/W | Break Address Mask B<br>Specifies bits masked in the break address of channel B specified by BARB (BAB31 to BAB0).<br>0: Break address BABn of channel B is included in the break condition<br>1: Break address BABn of channel B is masked and is not included in the break condition<br>Note: n = 31 to 0 |

### 7.3.8 Break Data Register B (BDRB)

BDRB is a 32-bit readable/writable register. The control bits CDB1 and CDB0 in BBRB select one of the two data buses for break condition B.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BDB31 | BDB30 | BDB29 | BDB28 | BDB27 | BDB26 | BDB25 | BDB24 | BDB23 | BDB22 | BDB21 | BDB20 | BDB19 | BDB18 | BDB17 | BDB16 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BDB15 | BDB14 | BDB13 | BDB12 | BDB11 | BDB10 | BDB9 | BDB8 | BDB7 | BDB6 | BDB5 | BDB4 | BDB3 | BDB2 | BDB1 | BDB0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | BDB31 to BDB0 | All 0 | R/W | Break Data Bit B<br>Stores data which specifies a break condition in channel B.<br>If the I bus is selected in BBRB, the break data on IDB is set in BDB31 to BDB0.<br>If the L bus is selected in BBRB, the break data on LDB is set in BDB31 to BDB0. |

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.

2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDRB as the break data.

### 7.3.9 Break Data Mask Register B (BDMRB)

BDMRB is a 32-bit readable/writable register. BDMRB specifies bits masked in the break data specified by BDRB.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BDMB31 | BDMB30 | BDMB29 | BDMB28 | BDMB27 | BDMB26 | BDMB25 | BDMB24 | BDMB23 | BDMB22 | BDMB21 | BDMB20 | BDMB19 | BDMB18 | BDMB17 | BDMB16 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BDMB15 | BDMB14 | BDMB13 | BDMB12 | BDMB11 | BDMB10 | BDMB9 | BDMB8 | BDMB7 | BDMB6 | BDMB5 | BDMB4 | BDMB3 | BDMB2 | BDMB1 | BDMB0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | BDMB31 to BDMB 0 | All 0 | R/W | Break Data Mask B<br>Specifies bits masked in the break data of channel B specified by BDRB (BDB31 to BDB0).<br>0: Break data BDBn of channel B is included in the break condition<br>1: Break data BDBn of channel B is masked and is not included in the break condition<br>Note: n = 31 to 0 |

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.

2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDMRB as the break mask data in BDRB.

### 7.3.10 Break Bus Cycle Register B (BBRB)

BBRB is a 16-bit readable/writable register, which specifies (1) bus master for I bus cycle, (2) L bus cycle or I bus cycle, (3) instruction fetch or data access, (4) read or write, and (5) operand size in the break conditions of channel B.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | CPB[2:0] | | | CDB[1:0] | | IDB[1:0] | | RWB[1:0] | | SZB[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/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 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 10 to 8 | CPB[2:0] | 000 | R/W | Bus Master Select B for I Bus<br>Select the bus master when the I bus is selected as the bus cycle of the channel B break condition. However, when the L bus is selected as the bus cycle, the setting of the CPB2 to CPB0 bits is disabled.<br>000: Condition comparison is not performed<br>xx1: The CPU cycle is included in the break condition<br>x1x: Setting prohibited<br>1xx: The DTC cycle is included in the break condition |
| 7, 6 | CDB[1:0] | 00 | R/W | L Bus Cycle/I Bus Cycle Select B<br>Select the L bus cycle or I bus cycle as the bus cycle of the channel B break condition.<br>00: Condition comparison is not performed<br>01: The break condition is the L bus cycle<br>10: The break condition is the I bus cycle<br>11: The break condition is the L bus cycle |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5, 4 | IDB[1:0] | 00 | R/W | Instruction Fetch/Data Access Select B<br>Select the instruction fetch cycle or data access cycle as the bus cycle of the channel B break condition.<br>00: Condition comparison is not performed<br>01: The break condition is the instruction fetch cycle<br>10: The break condition is the data access cycle<br>11: The break condition is the instruction fetch cycle or data access cycle |
| 3, 2 | RWB[1:0] | 00 | R/W | Read/Write Select B<br>Select the read cycle or write cycle as the bus cycle of the channel B break condition.<br>00: Condition comparison is not performed<br>01: The break condition is the read cycle<br>10: The break condition is the write cycle<br>11: The break condition is the read cycle or write cycle |
| 1, 0 | SZB[1:0] | 00 | R/W | Operand Size Select B<br>Select the operand size of the bus cycle for the channel B break condition.<br>00: The break condition does not include operand size<br>01: The break condition is byte access<br>10: The break condition is word access<br>11: The break condition is longword access<br>Note: When specifying the operand size, specify the size which matches the address boundary. |

[Legend]

x: Don't care.

### 7.3.11 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Specifies whether channels A and B conditions are used as two independent conditions or as the sequential condition.
2. Specifies whether a user break is set before or after instruction execution.
3. Specifies whether to include the number of execution times in channel B comparison conditions.
4. Specifies whether to include data bus in channels A and B comparison conditions.
5. Enables PC trace.
6. Selects the pulse width of the $\overline{\text{UBCTRG}}$ output.
7. Specifies whether to request a user break interrupt on a match of channels A and B comparison conditions.

BRCR is a 32-bit readable/writable register that has break conditions match flags and bits for setting a variety of break conditions.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | UTRGW[1:0] | | UBIDB | - | UBIDA | - |
| 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 | R/W | R |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SCM FCA | SCM FCB | SCM FDA | SCM FDB | PCTE | PCBA | - | - | DBEA | PCBB | DBEB | - | SEQ | - | - | ETBE |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W | R/W | R | R/W | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 22 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 21, 20 | UTRGW[1:0] | 00 | R/W | $\overline{\text{UBCTRG}}$ Output Pulse Width Select<br>Select the $\overline{\text{UBCTRG}}$ output pulse width when the break condition matches.<br>00: Setting prohibited<br>01: $\overline{\text{UBCTRG}}$ output pulse width is 3 to 4 $t_{Bcyc}$<br>10: $\overline{\text{UBCTRG}}$ output pulse width is 7 to 8 $t_{Bcyc}$<br>11: $\overline{\text{UBCTRG}}$ output pulse width is 15 to 16 $t_{Bcyc}$<br>Note: $t_{Bcyc}$ indicates the period of one cycle of the external bus clock (Bϕ = CK). |
| 19 | UBIDB | 0 | R/W | User Break Disable B<br>Enables or disables the user break interrupt request when the channel B break conditions are satisfied.<br>0: User break interrupt request is enabled when break conditions are satisfied<br>1: User break interrupt request is disabled when break conditions are satisfied |
| 18 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 17 | UBIDA | 0 | R/W | User Break Disable A<br>Enables or disables the user break interrupt request when the channel A break conditions are satisfied.<br>0: User break interrupt request is enabled when break conditions are satisfied<br>1: User break interrupt request is disabled when break conditions are satisfied |
| 16 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | SCMFCA | 0 | R/W | L Bus Cycle Condition Match Flag A<br>When the L bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1. In order to clear this flag, write 0 into this bit.<br>0: The L bus cycle condition for channel A does not match<br>1: The L bus cycle condition for channel A matches |
| 14 | SCMFCB | 0 | R/W | L Bus Cycle Condition Match Flag B<br>When the L bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1. In order to clear this flag, write 0 into this bit.<br>0: The L bus cycle condition for channel B does not match<br>1: The L bus cycle condition for channel B matches |
| 13 | SCMFDA | 0 | R/W | I Bus Cycle Condition Match Flag A<br>When the I bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1. In order to clear this flag, write 0 into this bit.<br>0: The I bus cycle condition for channel A does not match<br>1: The I bus cycle condition for channel A matches |
| 12 | SCMFDB | 0 | R/W | I Bus Cycle Condition Match Flag B<br>When the I bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1. In order to clear this flag, write 0 into this bit.<br>0: The I bus cycle condition for channel B does not match<br>1: The I bus cycle condition for channel B matches |
| 11 | PCTE | 0 | R/W | PC Trace Enable<br>0: Disables PC trace<br>1: Enables PC trace |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 | PCBA | 0 | R/W | PC Break Select A<br>Selects the break timing of the instruction fetch cycle for channel A as before or after instruction execution.<br>0: PC break of channel A is set before instruction execution<br>1: PC break of channel A is set after instruction execution |
| 9, 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 7 | DBEA | 0 | R/W | Data Break Enable A<br>Selects whether or not the data bus condition is included in the break condition of channel A.<br>0: No data bus condition is included in the condition of channel A<br>1: The data bus condition is included in the condition of channel A |
| 6 | PCBB | 0 | R/W | PC Break Select B<br>Selects the break timing of the instruction fetch cycle for channel B as before or after instruction execution.<br>0: PC break of channel B is set before instruction execution<br>1: PC break of channel B is set after instruction execution |
| 5 | DBEB | 0 | R/W | Data Break Enable B<br>Selects whether or not the data bus condition is included in the break condition of channel B.<br>0: No data bus condition is included in the condition of channel B<br>1: The data bus condition is included in the condition of channel B |
| 4 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | SEQ | 0 | R/W | Sequence Condition Select<br>Selects two conditions of channels A and B as independent or sequential conditions.<br>0: Channels A and B are compared under independent conditions<br>1: Channels A and B are compared under sequential conditions (channel A, then channel B) |
| 2, 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | ETBE | 0 | R/W | Number of Execution Times Break Enable<br>Enables the execution-times break condition only on channel B. If this bit is 1 (break enable), a user break is issued when the number of break conditions matches with the number of execution times that is specified by BETR.<br>0: The execution-times break condition is disabled on channel B<br>1: The execution-times break condition is enabled on channel B |

### 7.3.12 Execution Times Break Register (BETR)

BETR is a 16-bit readable/writable register. When the execution-times break condition of channel B is enabled, this register specifies the number of execution times to make the break. The maximum number is $2^{12} - 1$ times. When a break condition is satisfied, it decreases BETR. A user break interrupt is requested when the break condition is satisfied after BETR becomes H'0001.

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 12 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 11 to 0 | BET[11:0] | All 0 | R/W | Number of Execution Times |

### 7.3.13 Branch Source Register (BRSR)

BRSR is a 32-bit read-only register. BRSR stores bits 27 to 0 in the address of the branch source instruction. BRSR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRSR is read, the setting to enable PC trace is made, or BRSR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The four BRSR registers have a queue structure and a stored register is shifted at every branch.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SVF | - | - | - | BSA27 | BSA26 | BSA25 | BSA24 | BSA23 | BSA22 | BSA21 | BSA20 | BSA19 | BSA18 | BSA17 | BSA16 |
| Initial value: | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSA15 | BSA14 | BSA13 | BSA12 | BSA11 | BSA10 | BSA9 | BSA8 | BSA7 | BSA6 | BSA5 | BSA4 | BSA3 | BSA2 | BSA1 | BSA0 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | SVF | 0 | R | BRSR Valid Flag<br>Indicates whether the branch source address is stored. This flag bit is set to 1 when a branch occurs. This flag is cleared to 0 when BRSR is read, the setting to enable PC trace is made, or BRSR is initialized by a power-on reset.<br>0: The value of BRSR register is invalid<br>1: The value of BRSR register is valid |
| 30 to 28 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 27 to 0 | BSA27 to BSA0 | Undefined | R | Branch Source Address<br>Store bits 27 to 0 of the branch source address. |

### 7.3.14 Branch Destination Register (BRDR)

BRDR is a 32-bit read-only register. BRDR stores bits 27 to 0 in the address of the branch destination instruction. BRDR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRDR is read, the setting to enable PC trace is made, or BRDR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The four BRDR registers have a queue structure and a stored register is shifted at every branch.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DVF | - | - | - | BDA27 | BDA26 | BDA25 | BDA24 | BDA23 | BDA22 | BDA21 | BDA20 | BDA19 | BDA18 | BDA17 | BDA16 |
| Initial value: | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BDA15 | BDA14 | BDA13 | BDA12 | BDA11 | BDA10 | BDA9 | BDA8 | BDA7 | BDA6 | BDA5 | BDA4 | BDA3 | BDA2 | BDA1 | BDA0 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | DVF | 0 | R | BRDR Valid Flag<br>Indicates whether a branch destination address is stored. This flag bit is set to 1 when a branch occurs. This flag is cleared to 0 when BRDR is read, the setting to enable PC trace is made, or BRDR is initialized by a power-on reset.<br>0: The value of BRDR register is invalid<br>1: The value of BRDR register is valid |
| 30 to 28 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 27 to 0 | BDA27 to BDA0 | Undefined | R | Branch Destination Address<br>Store bits 27 to 0 of the branch destination address. |

## 7.4 Operation

### 7.4.1 Flow of the User Break Operation

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

1. The break addresses are set in the break address registers (BARA or BARB). The masked addresses are set in the break address mask registers (BAMRA or BAMRB). The break data is set in the break data register (BDRA or BDRB). The masked data is set in the break data mask register (BDMRA or BDMRB). The bus break conditions are set in the break bus cycle registers (BBRA or BBRB). Three groups of BBRA or BBRB (L 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 with B'00. The respective conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBRA or BBRB.
2. When the break conditions are satisfied, the UBC issues a user break interrupt request to the CPU and sets the L bus condition match flag (SCMFCA or SCMFCB) and the I bus condition match flag (SCMFDA or SCMFDB) for the appropriate channel.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMFDB) can be used to check if the set conditions match or not. The matching of the conditions sets flags, but they are not reset. Before using them again, 0 must first be written to them and then reset flags.
4. There is a possibility that matches of the break conditions set in channels A and B occur almost at the same time. In this case, only one user break interrupt request may be sent to the CPU with both of the two condition match flags set.
5. When selecting the I bus as the break condition, note the following:
   - The CPU and DTC are connected to the I bus. The UBC monitors bus cycles generated by all bus masters that are selected by the CPA2 to CPA0 bits in BBRA or the CPB2 to CPB0 bits in BBRB, and compares for a condition match.
   - I bus cycles resulting from instruction fetches on the L bus by the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles are defined as data access cycles.
   - The DTC only issue data access cycles for I bus cycles.
   - If a break condition is specified for the I bus, even when the condition matches in an I bus cycle resulting from an instruction executed by the CPU, at which instruction the break is to be accepted cannot be clearly defined.

### 7.4.2 User Break on Instruction Fetch Cycle

1. When L bus/instruction fetch/read/word, longword, or the operand size is not included is set in the break bus cycle register (BBRA or BBRB), the break condition becomes the L bus instruction fetch cycle. Whether a user break is set before or after the execution of the instruction can then be selected with the PCBA or PCBB bit in the break control register (BRCR) for the corresponding channel. If an instruction fetch cycle is set as a break condition, clear LSB in the break address register (BARA or BARB) to 0. A user break cannot be generated as long as this bit is set to 1.
2. If the break condition matches when a user break on instruction fetch is specified so that the a break is generated before the execution of the instruction, the user break is generated at the point when it has become deterministic that the instruction will be executed after it is fetched. This means this feature cannot be used on instructions fetched by overrun (instructions fetched at a branch or during an interrupt transition, but not executed). When this kind of break condition is set for the delay slot of a delayed branch instruction, a user break is generated prior to execution of the delayed branch instruction.

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

3. When the break condition is specified so that a user break is generated after execution of the instruction, the instruction that has met the break condition is executed and then the user break is generated before the next instruction is executed. As with pre-execution user breaks, this cannot be used with overrun fetch instructions. When this kind of break condition is set for a delayed branch instruction and its delay slot, a user break is not generated until the processing jumps to the first instruction at the branch destination.
4. When an instruction fetch cycle is set, the break data register (BDRA or BDRB) is ignored. Therefore, break data cannot be set for the user break of the instruction fetch cycle.
5. If the I bus is set as the condition for a user break on instruction fetch cycle, the I bus is monitored for instruction fetch cycles to detect condition match. For details, see 5 in section 7.4.1, Flow of the User Break Operation.

### 7.4.3 Break on Data Access Cycle

1. If the L bus is specified as a break condition for data access break, condition comparison is performed for the address (and data) accessed by the executed instructions, and a user break is generated 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 that are issued on the I bus by all bus masters including the CPU, and a user break is generated if the condition is satisfied. For details on the CPU bus cycles issued on the I bus, see 5 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 (BARA or BARB), 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 of the break bus cycle register (BBRA or BBRB). When data values are included in break conditions, a user break is generated when the address conditions and data conditions both match. To specify byte data for this case, set the same data in two bytes at bits 15 to 8 and bits 7 to 0 of the break data register (BDRA or BDRB) and break data mask register (BDMRA or BDMRB). When word or byte is set, bits 31 to 16 of BDRA or BDRB and BDMRA or BDMRB are ignored.
4. If the L bus is selected, a user break is generated on ending execution of the instruction that matches the break condition, and immediately before the next instruction is executed. However, when data is also specified as the break condition, the break may occur on ending execution of the instruction following the instruction that matches the break condition. When the I bus is selected, the instruction at which the user break is generated cannot be determined.

When this kind of break occurs at a delayed branch instruction or its delay slot, the break may not actually take place until the processing jumps to the first instruction at the branch destination.

### 7.4.4 Sequential Break

1. By setting the SEQ bit in BRCR to 1, the sequential break is issued when a channel B break condition matches after a channel A break condition matches. A user break is not generated even if a channel B break condition matches before a channel A break condition matches. When channels A and B conditions match at the same time, the sequential break is not issued. To clear the channel A condition match when a channel A condition match has occurred but a channel B condition match has not yet occurred when a sequential break has been specified, clear the SEQ bit in BRCR and channel A condition match flag to 0 by writing a 0 to them.
2. In sequential break specification, the L or I bus can be selected and the execution times break condition can be also specified. For example, when the execution times break condition is specified, the break condition is satisfied when a channel B condition matches with BETR = H'0001 after a channel A condition has matched.

### 7.4.5 Value of Saved Program Counter

When a user break occurs, the address of the instruction from where execution is to be resumed is saved in the stack, and the exception handling state is entered. If the L bus is specified as the break condition, the instruction at which the user break should occur can be clearly determined (except for when data is included in the break condition). If the I bus is specified as a break condition, the instruction at which the user break should occur cannot be clearly determined.

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

   The address of the instruction that matched the break condition is saved in the stack. The instruction that matched the condition is not executed, and the user break occurs before it. However when a delay slot instruction matches the condition, the address of the delayed branch instruction is saved in the stack.
2. When 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 in 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, these instructions are executed, and the branch destination address is saved in the stack.

3. When data access (address only) is specified as a break condition:

   The address of the instruction immediately after the instruction that matched the break condition is saved in the stack. The instruction that matches the condition is executed, and the user break occurs before the next instruction is executed. However when a delay slot instruction matches the condition, the branch destination address is saved in the stack.
4. When data access (address + data) is specified as a break condition:

   When a data value is added to the break conditions, the address of an instruction that is within two instructions of the instruction that matched the break condition is saved in the stack. At which instruction the user break occurs cannot be determined accurately.

   When a delay slot instruction matches the condition, the branch destination address is saved in the stack. If the instruction following the instruction that matches the break condition is a branch instruction, the break may occur after the branch instruction or delay slot has finished. In this case, the branch destination address is saved in the stack.

### 7.4.6 PC Trace

1. Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, and interrupt exception) is generated, the branch source address and branch destination address are stored in BRSR and BRDR, respectively.
2. The values stored in BRSR and BRDR are as given below due to the kind of branch.
   - If a branch occurs due to a branch instruction, the address of the branch instruction is saved in BRSR and the address of the branch destination instruction is saved in BRDR.
   - If a branch occurs due to an interrupt or exception, the value saved in stack due to exception occurrence is saved in BRSR and the start address of the exception handling routine is saved in BRDR.
3. BRSR and BRDR have four pairs of queue structures. The top of queues is read first when the address stored in the PC trace register is read. BRSR and BRDR share the read pointer. Read BRSR and BRDR in order, the queue only shifts after BRDR is read. After switching the PCTE bit (in BRCR) off and on, the values in the queues are invalid.
4. Since four pairs of queue are shared with the AUD, set the PCTE bit in BRCR to 1 after setting the MSTP25 bit in STBCR5 to 0 and the AUDSRST bit in STBCR6 to 1. This setting is necessary even though this LSI does not have the AUD function.
5. A status of FIFO is initialized by a power-on reset, manual reset, or AUD software reset. When the status of FIFO is initialized by a manual reset or an AUD software reset, clear the PCTE bit in the BRCR register to 0 once, set the PCTE bit to 1, and then the PC trace can start.

### 7.4.7 Usage Examples

**Break Condition Specified for L Bus Instruction Fetch Cycle:**

(Example 1-1)

- Register specifications

  BARA = H'00000404, BAMRA = H'00000000, BBRA = H'0054, BDRA = H'00000000, BDMRA = H'00000000, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000400

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

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

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

  <Channel B>

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

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

  Bus cycle: L 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

  BARA = H'00037226, BAMRA = H'00000000, BBRA = H'0056, BDRA = H'00000000, BDMRA = H'00000000, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008

  Specified conditions: Channel A/channel B sequential mode

  <Channel A>

  Address: H'00037226, Address mask: H'00000000

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

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

<Channel B>

Address: H'0003722E, Address mask: H'00000000

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

Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

After an instruction with address H'00037226 is executed, a user break occurs before an instruction with address H'0003722E is executed.

(Example 1-3)

- Register specifications

  BARA = H'00027128, BAMRA = H'00000000, BBRA = H'005A, BDRA = H'00000000, BDMRA = H'00000000, BARB = H'00031415, BAMRB = H'00000000, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000000

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

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

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

  <Channel B>

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

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

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

  On channel A, no user break occurs since instruction fetch is not a write cycle. On channel B, no user break occurs since instruction fetch is performed for an even address.

(Example 1-4)

- Register specifications

  BARA = H'00037226, BAMRA = H'00000000, BBRA = H'005A, BDRA = H'00000000, BDMRA = H'00000000, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008

  Specified conditions: Channel A/channel B sequential mode

  <Channel A>

  Address: H'00037226, Address mask: H'00000000

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

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

<Channel B>

Address: H'0003722E, Address mask: H'00000000

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

Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

Since instruction fetch is not a write cycle on channel A, a sequential condition does not match. Therefore, no user break occurs.

(Example 1-5)

- Register specifications

  BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BDRA = H'00000000, BDMRA = H'00000000, BARB = H'00001000, BAMRB = H'00000000, BBRB = H'0057, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000001, BETR = H'0005

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00000500, Address mask: H'00000000

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

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

  The number of execution-times break enable (5 times)

  <Channel B>

  Address: H'00001000, Address mask: H'00000000

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

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

  On channel A, a user break occurs after the instruction of address H'00000500 is executed four times and before the fifth time.

  On channel B, a user break occurs before an instruction of address H'00001000 is executed.

(Example 1-6)

- Register specifications

  BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BDRA = H'00000000, BDMRA = H'00000000, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000400

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

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

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

<Channel B>

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

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

Bus cycle: L 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.

**Break Condition Specified for L Bus Data Access Cycle:**

(Example 2-1)

- Register specifications

  BARA = H'00123456, BAMRA = H'00000000, BBRA = H'0064, BDRA = H'12345678, BDMRA = H'FFFFFFFF, BARB = H'000ABCDE, BAMRB = H'000000FF, BBRB = H'006A, BDRB = H'0000A512, BDMRB = H'00000000, BRCR = H'00000080

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

  Data: H'12345678, Data mask: H'FFFFFFFF

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

  <Channel B>

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

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

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

  On channel A, a user break occurs with longword read from address H'00123454, word read from address H'00123456, or byte read from address H'00123456. On channel B, a user break occurs when word H'A512 is written in addresses H'000ABC00 to H'000ABCFE.

**Break Condition Specified for I Bus Data Access Cycle:**

(Example 3-1)

- Register specifications

  BARA = H'00314154, BAMRA = H'00000000, BBRA = H'0194, BDRA = H'12345678, BDMRA = H'FFFFFFFF, BARB = H'00055555, BAMRB = H'00000000, BBRB = H'01A9, BDRB = H'00007878, BDMRB = H'00000F0F, BRCR = H'00000080

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00314154, Address mask: H'00000000

  Data: H'12345678, Data mask: H'FFFFFFFF

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

  <Channel B>

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

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

  Bus cycle: I bus (CPU cycle)/data access/write/byte

  On channel A, a user break occurs when instruction fetch is performed for address H'00314156 in the external memory space.

  On channel B, a user break occurs when byte data H'7x is written in address H'00055555 in the external memory space by the CPU.

## 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 user 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. UBC cannot monitor access to the L bus and I bus in the same channel.
3. Note on specification of sequential break:

   A condition match occurs when a B-channel match occurs in a bus cycle after an A-channel match occurs in another bus cycle in sequential break setting. Therefore, no user break occurs if a bus cycle in which an A-channel match and a channel B match occur simultaneously is set.
4. When a user break and another exception occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 5.1 in section 5, Exception Handling. If an exception with higher priority occurs, the user break is not generated.
   - Pre-execution break has the highest priority.
   - When a post-execution break or data access break occurs simultaneously with a re-execution-type exception (including pre-execution break) that has higher priority, the re-execution-type exception is accepted, and the condition match flag is not set (see the exception in the following note). The user break will occur and the condition match flag will be set only after the exception source of the re-execution-type exception has been cleared by the exception handling routine and re-execution of the same instruction has ended.
   - When a post-execution break or data access break occurs simultaneously with a completion-type exception (TRAPA) that has higher priority, a user break does not occur but the condition match flag is set.
5. Note the following exception for the above note.

   If a post-execution break or data access break is satisfied by an instruction that generates a CPU address error by data access, the CPU address error takes priority over the user break. Note that the UBC condition match flag is set in this case.
6. Note the following when a user break occurs in a delay slot.

   If a pre-execution break is set at the delay slot instruction of the RTE instruction, the user break does not occur until the branch destination of the RTE instruction.

7. 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.
8. Do not set a post-execution break at a SLEEP instruction or a branch instruction for which a SLEEP instruction is placed in the delay slot. In addition, do not set a data access break at a SLEEP instruction or one or two instructions before a SLEEP instruction.
9. The UBC cannot detect external space accesses by the CPU on the I bus correctly when the DTC is in operation. Select all bus masters when determining an external space access on the I bus in the above condition. In this case, conditions for an identified bus master cannot be set. However, if the bus master can be inferred from the data value, the bus master can be inferred by including the data as a match condition.

# Section 8 Data Transfer Controller (DTC)

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

## 8.1 Features

- Transfer possible over any number of channels:
- Chain transfer
  Multiple rounds of data transfer is executed in response to a single activation source
  Chain transfer is only possible after data transfer has been done for the specified number of times (i.e. when the transfer counter is 0)
- Three transfer modes
  Normal/repeat/block transfer modes selectable
  Transfer source and destination addresses can be selected from increment/decrement/fixed
- The transfer source and destination addresses can be specified by 32 bits to select a 4-Gbyte address space directly
- Size of data for data transfer can be specified as byte, word, or longword
- A CPU interrupt can be requested for the interrupt that activated the DTC
  A CPU interrupt can be requested after one data transfer completion
  A CPU interrupt can be requested after the specified data transfer completion
- Read skip of the transfer information specifiable
- Writeback skip executed for the fixed transfer source and destination addresses
- Module stop mode specifiable
- Short address mode specifiable
- Bus release timing selectable from five types
- Priority of the DTC activation selectable from two types

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

Note: * When the transfer information is stored in the on-chip RAM, the RAME bit in RAMCR must be set to 1.

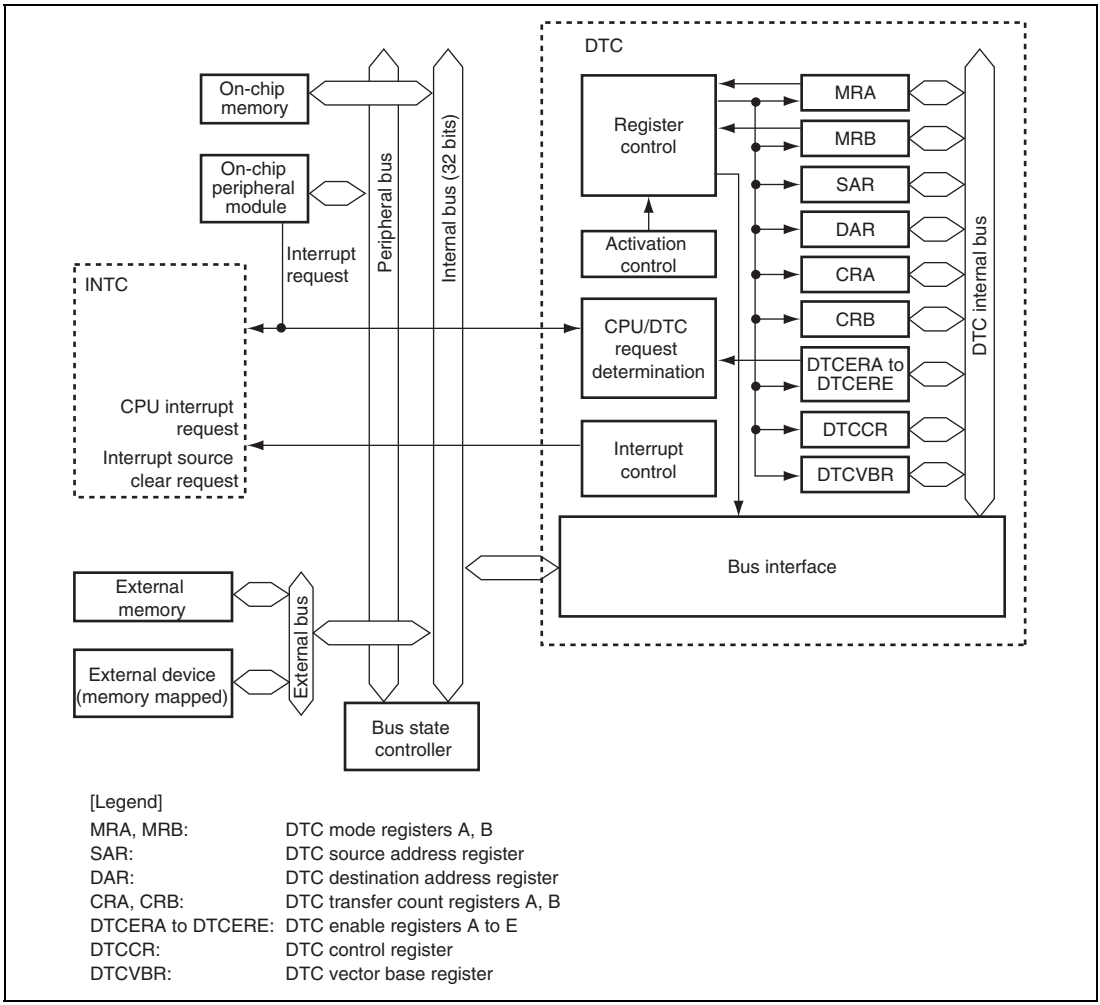



**Figure 8.1 Block Diagram of DTC**

## 8.2 Register Descriptions

DTC has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 25, List of Registers.

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

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

**Table 8.1 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| DTC enable register A | DTCERA | R/W | H'0000 | H'FFFFCC80 | 8, 16 |
| DTC enable register B | DTCERB | R/W | H'0000 | H'FFFFCC82 | 8, 16 |
| DTC enable register C | DTCERC | R/W | H'0000 | H'FFFFCC84 | 8, 16 |
| DTC enable register D | DTCERD | R/W | H'0000 | H'FFFFCC86 | 8, 16 |
| DTC enable register E | DTCERE | R/W | H'0000 | H'FFFFCC88 | 8, 16 |
| DTC control register | DTCCR | R/W | H'00 | H'FFFFCC90 | 8 |
| DTC vector base register | DTCVBR | R/W | H'00000000 | H'FFFFCC94 | 8, 16, 32 |
| Bus function extending register | BSCEHR | R/W | H'0000 | H'FFFFE89A | 8, 16 |

### 8.2.1 DTC Mode Register A (MRA)

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

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

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

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

[Legend]
x: Don't care

### 8.2.2 DTC Mode Register B (MRB)

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CHNE | Undefined | — | DTC Chain Transfer Enable<br>Specifies the chain transfer. For details, see section 8.5.6, Chain Transfer. The chain transfer condition is selected by the CHNS bit.<br>0: Disables the chain transfer<br>1: Enables the chain transfer |
| 6 | CHNS | Undefined | — | DTC Chain Transfer Select<br>Specifies the chain transfer condition. If the following transfer is a chain transfer, the completion check of the specified transfer count is not performed and activation source flag or DTCER is not cleared.<br>0: Chain transfer every time<br>1: Chain transfer only when transfer counter = 0 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | DISEL | Undefined | — | DTC Interrupt Select<br>When this bit is set to 1, an interrupt request is generated to the CPU every time a data transfer or a block transfer ends. When this bit is set to 0, a CPU interrupt request is only generated when the specified number of data transfers end. |
| 4 | DTS | Undefined | — | DTC Transfer Mode Select<br>Specifies either the source or destination as repeat or block area during repeat or block transfer mode.<br>0: Specifies the destination as repeat or block area<br>1: Specifies the source as repeat or block area |
| 3, 2 | DM[1:0] | Undefined | — | Destination Address Mode 1 and 0<br>Specify a DAR operation after a data transfer.<br>0x: DAR is fixed<br>(DAR writeback is skipped)<br>10: DAR is incremented after a transfer<br>(by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10)<br>11: SAR is decremented after a transfer<br>(by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10) |
| 1, 0 | — | Undefined | — | Reserved<br>The write value should always be 0. |

[Legend]
x: Don't care

### 8.2.3 DTC Source Address Register (SAR)

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

SAR cannot be accessed directly from the CPU.

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

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

*: Undefined

### 8.2.4 DTC Destination Address Register (DAR)

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

DAR cannot be accessed directly from the CPU.

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

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

*: Undefined

### 8.2.5 DTC Transfer Count Register A (CRA)

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

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

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

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

CRA cannot be accessed directly from the CPU.

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

* : Undefined

### 8.2.6 DTC Transfer Count Register B (CRB)

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

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

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

* : Undefined

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

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

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

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

### 8.2.8 DTC Control Register (DTCCR)

DTCCR specifies transfer information read skip.

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

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2, 1 | — | All 0 | R | Reserved<br>These are read-only bits and cannot be modified. |
| 0 | ERR | 0 | R/(W)* | Transfer Stop Flag<br>Indicates that the DTC address error or NMI interrupt request has occurred. If a DTC address error or NMI interrupt occurs while the DTC is active, address error handling or NMI interrupt handling processing is executed after the DTC has released the bus mastership. The DTC stops in the transfer information writing state after transferring data.<br>0: No interrupt occurs<br>1: An interrupt occurs<br>[Clearing condition]<br>• When writing 0 after reading 1 |

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

### 8.2.9 DTC Vector Base Register (DTCVBR)

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

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

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

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

### 8.2.10 Bus Function Extending Register (BSCEHR)

BSCEHR is a 16-bit register that specifies the timing of bus release by the DTC and other functions. This register can be used to give higher priority to the transfer by the DTC and configure the functions that can reduce the number of cycles over which the DTC is active. For more details, see section 9.4.4, Bus Function Extending Register (BSCEHR).

## 8.3 Activation Sources

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

## 8.4 Location of Transfer Information and DTC Vector Table

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

Only in the case where all transfer sources/transfer destinations are in on-chip RAM and on-chip peripheral modules, short address mode can be selected by setting the DTSA bit in the bus function extending register (BSCEHR) to 1 (see section 9.4.4, Bus Function Extending Register (BSCEHR)).
Normally, four longwords of transfer information has to be read. But if short address mode is selected, the size of transfer information is reduced to three longwords, which can shorten the period over which the DTC is active.

The DTC reads the start address of the transfer information from the vector table for every activation source and reads the transfer information from this start address. Figure 8.3 shows correspondences between the DTC vector table and transfer information.

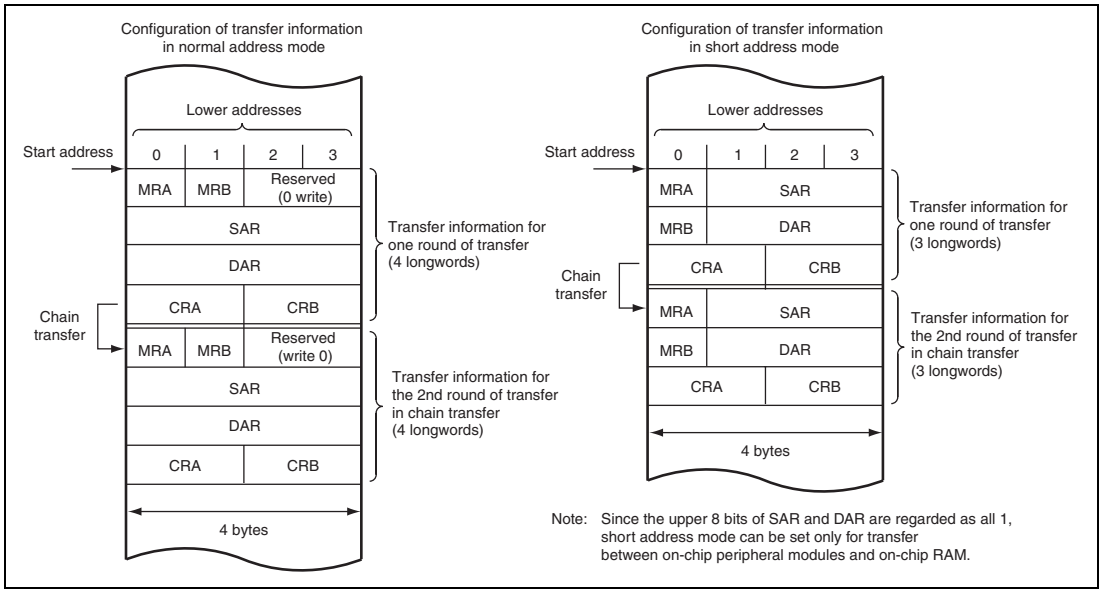



**Figure 8.2 Transfer Information on Data Area**

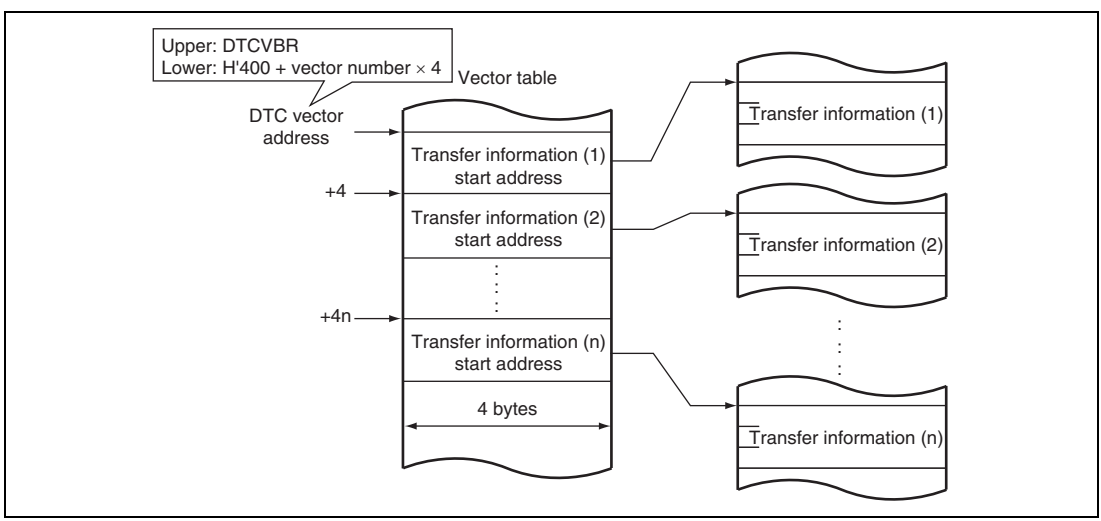



**Figure 8.3 Correspondence between DTC Vector Address and Transfer Information**

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

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

| Origin of Activation Source | Activation Source | Vector Number | DTC Vector Address Offset | DTCE*[1] | Transfer Source | Transfer Destination | Priority |
|---|---|---|---|---|---|---|---|
| External pin | IRQ0 | 64 | H'500 | DTCERA15 | Arbitrary*[2] | Arbitrary*[2] | High |
| | IRQ1 | 65 | H'504 | DTCERA14 | Arbitrary*[2] | Arbitrary*[2] | ↑ |
| | IRQ2 | 66 | H'508 | DTCERA13 | Arbitrary*[2] | Arbitrary*[2] | |
| | IRQ3 | 67 | H'50C | DTCERA12 | Arbitrary*[2] | Arbitrary*[2] | |
| MTU2_0 | TGIA_0 | 88 | H'560 | DTCERB15 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIB_0 | 89 | H'564 | DTCERB14 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIC_0 | 90 | H'568 | DTCERB13 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGID_0 | 91 | H'56C | DTCERB12 | Arbitrary*[2] | Arbitrary*[2] | |
| MTU2_1 | TGIA_1 | 96 | H'580 | DTCERB11 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIB_1 | 97 | H'584 | DTCERB10 | Arbitrary*[2] | Arbitrary*[2] | |
| MTU2_2 | TGIA_2 | 104 | H'5A0 | DTCERB9 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIB_2 | 105 | H'5A4 | DTCERB8 | Arbitrary*[2] | Arbitrary*[2] | |
| MTU2_3 | TGIA_3 | 112 | H'5C0 | DTCERB7 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIB_3 | 113 | H'5C4 | DTCERB6 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIC_3 | 114 | H'5C8 | DTCERB5 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGID_3 | 115 | H'5CC | DTCERB4 | Arbitrary*[2] | Arbitrary*[2] | |
| MTU2_4 | TGIA_4 | 120 | H'5E0 | DTCERB3 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIB_4 | 121 | H'5E4 | DTCERB2 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIC_4 | 122 | H'5E8 | DTCERB1 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGID_4 | 123 | H'5EC | DTCERB0 | Arbitrary*[2] | Arbitrary*[2] | |
| | TCIV_4 | 124 | H'5F0 | DTCERC15 | Arbitrary*[2] | Arbitrary*[2] | |
| MTU2_5 | TGIU_5 | 128 | H'600 | DTCERC14 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIV_5 | 129 | H'604 | DTCERC13 | Arbitrary*[2] | Arbitrary*[2] | ↓ |
| | TGIW_5 | 130 | H'608 | DTCERC12 | Arbitrary*[2] | Arbitrary*[2] | Low |

| Origin of Activation Source | Activation Source | Vector Number | DTC Vector Address Offset | DTCE*[1] | Transfer Source | Transfer Destination | Priority |
|---|---|---|---|---|---|---|---|
| MTU2S_3 | TGIA_3S | 160 | H'680 | DTCERC3 | Arbitrary*[2] | Arbitrary*[2] | High |
| | TGIB_3S | 161 | H'684 | DTCERC2 | Arbitrary*[2] | Arbitrary*[2] | ↑ |
| | TGIC_3S | 162 | H'688 | DTCERC1 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGID_3S | 163 | H'68C | DTCERC0 | Arbitrary*[2] | Arbitrary*[2] | |
| MTU2S_4 | TGIA_4S | 168 | H'6A0 | DTCERD15 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIB_4S | 169 | H'6A4 | DTCERD14 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIC_4S | 170 | H'6A8 | DTCERD13 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGID_4S | 171 | H'6AC | DTCERD12 | Arbitrary*[2] | Arbitrary*[2] | |
| | TCIV_4S | 172 | H'6B0 | DTCERD11 | Arbitrary*[2] | Arbitrary*[2] | |
| MTU2S_5 | TGIU_5S | 176 | H'6C0 | DTCERD10 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIV_5S | 177 | H'6C4 | DTCERD9 | Arbitrary*[2] | Arbitrary*[2] | |
| | TGIW_5S | 178 | H'6C8 | DTCERD8 | Arbitrary*[2] | Arbitrary*[2] | |
| CMT_0 | CMI_0 | 184 | H'6E0 | DTCERD7 | Arbitrary*[2] | Arbitrary*[2] | |
| CMT_1 | CMI_1 | 188 | H'6F0 | DTCERD6 | Arbitrary*[2] | Arbitrary*[2] | |
| A/D_0 | ADI_3 | 208 | H'740 | DTCERD2 | ADDR0 to ADDR7 | Arbitrary*[2] | |
| A/D_1 | ADI_4 | 212 | H'750 | DTCERD1 | ADDR8 to ADDR15 | Arbitrary*[2] | |
| SCI_0 | RXI_0 | 217 | H'764 | DTCERE15 | SCRDR_0 | Arbitrary*[2] | |
| | TXI_0 | 218 | H'768 | DTCERE14 | Arbitrary*[2] | SCTDR_0 | |
| SCI_1 | RXI_1 | 221 | H'774 | DTCERE13 | SCRDR_1 | Arbitrary*[2] | |
| | TXI_1 | 222 | H'778 | DTCERE12 | Arbitrary*[2] | SCTDR_1 | |
| SCI_2 | RXI_2 | 225 | H'784 | DTCERE11 | SCRDR_2 | Arbitrary*[2] | |
| | TXI_2 | 226 | H'788 | DTCERE10 | Arbitrary*[2] | SCTDR_2 | |
| SSU | SSRXI | 233 | H'7A4 | DTCERE7 | SSRDR0 to SSRDR3 | Arbitrary*[2] | |
| | SSTXI | 234 | H'7A8 | DTCERE6 | Arbitrary*[2] | SSTDR0 to SSTDR3 | |
| $I^2C2$ | IITXI | 238 | H'7B8 | DTCERE5 | Arbitrary*[2] | ICDRT | |
| | IIRXI | 239 | H'7BC | DTCERE4 | ICDRR | Arbitrary*[2] | |
| RCAN-ET_0 | RM0_0 | 242 | H'7C8 | DTCERE3 | CONTROL0H to CONTROL1L*[3] | Arbitrary*[2] | Low |

Notes: 1. The DTCE bits with no corresponding interrupt are reserved, and the write value should always be 0. To leave software standby mode with an interrupt, write 0 to the corresponding DTCE bit.

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

3. Read to a message control field 1 (CONTROL1) in mailbox 0 by using a block transfer mode or etc.

## 8.5 Operation

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

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

Table 8.3 shows the DTC transfer modes.

**Table 8.3 DTC Transfer Modes**

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

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

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

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

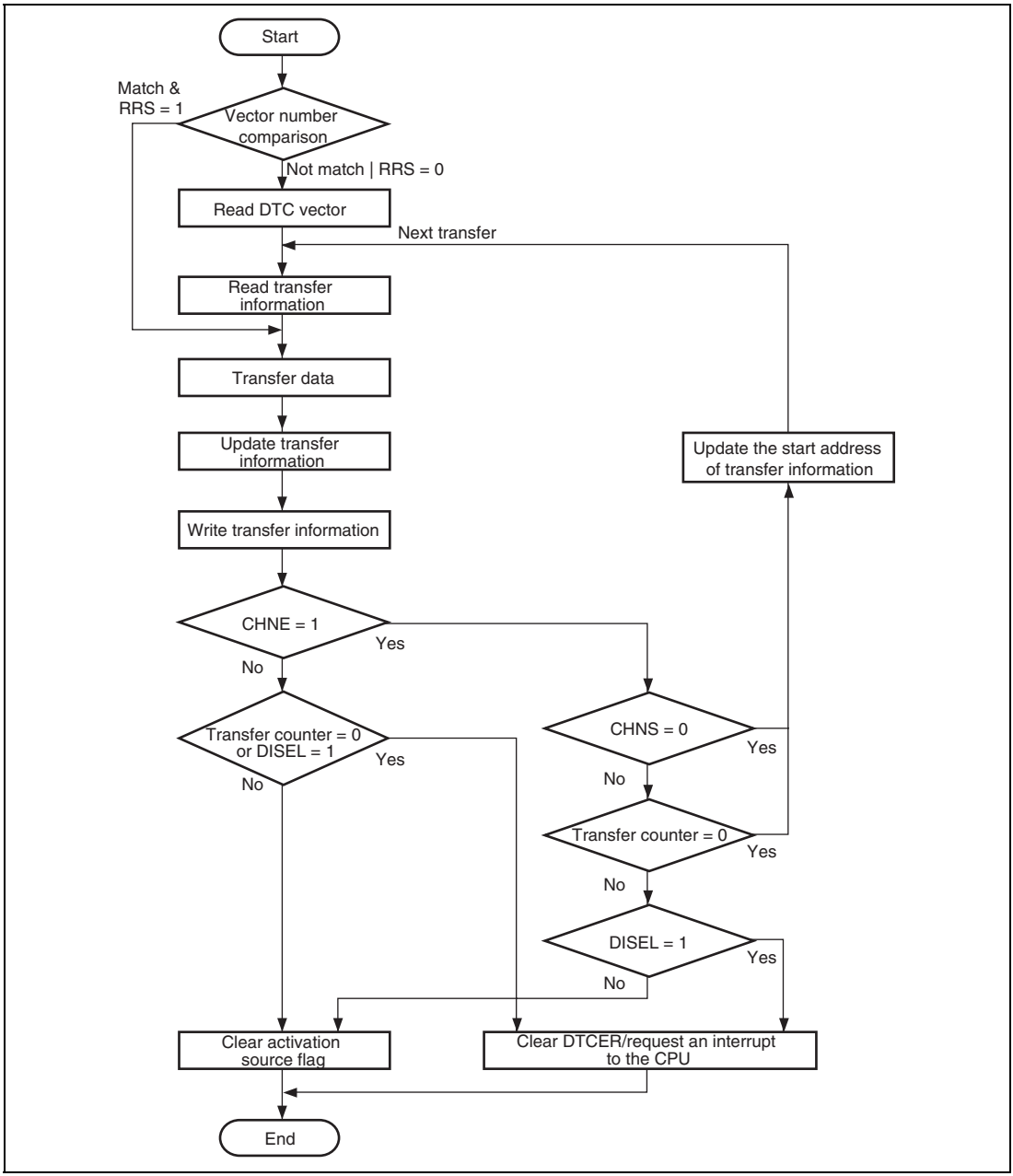



**Figure 8.4 Flowchart of DTC Operation**

**Table 8.4 DTC Transfer Conditions (Chain Transfer Conditions Included)**

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

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

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

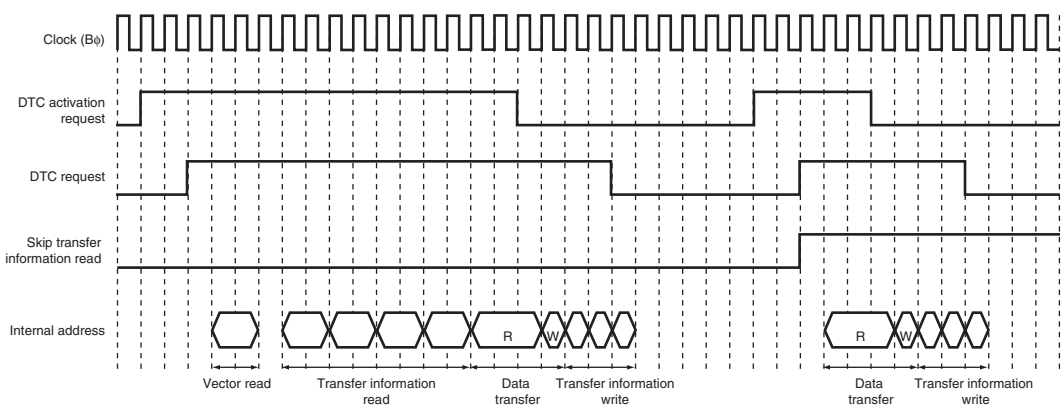
Notes: 1. CRA in normal mode transfer, CRAL in repeat transfer mode, or CRB in block transfer mode

2. When the contents of the CRAH is written to the CRAL in repeat transfer mode

### 8.5.1 Transfer Information Read Skip Function

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

To modify the vector table and transfer information, temporarily clear the RRS bit to 0, modify the vector table and transfer information, and then set the RRS bit to 1 again. When the RRS bit is cleared to 0, the stored vector number is deleted, and the updated vector table and transfer information are read at the next activation.
If the DTPR bit in the bus function extending register (BSCEHR) is set to 1, this function is always disabled.




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

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

### 8.5.2 Transfer Information Writeback Skip Function

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

**Table 8.5 Transfer Information Writeback Skip Condition and Writeback Skipped Registers**

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

### 8.5.3 Normal Transfer Mode

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

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

**Table 8.6 Register Function in Normal Transfer Mode**

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

Note: * Transfer information writeback is skipped.

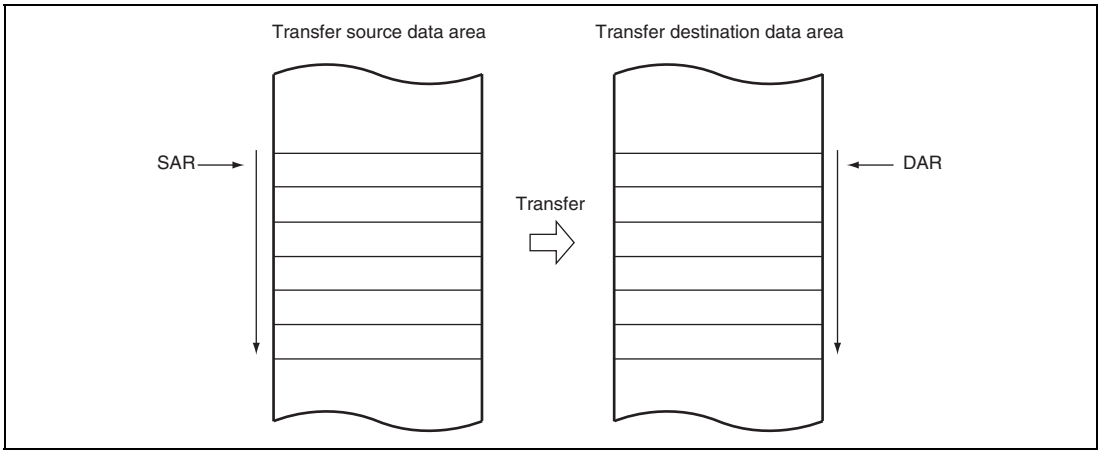



**Figure 8.6 Memory Map in Normal Transfer Mode**

### 8.5.4 Repeat Transfer Mode

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

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

**Table 8.7 Register Function in Repeat Transfer Mode**

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

Note: * Transfer information writeback is skipped.

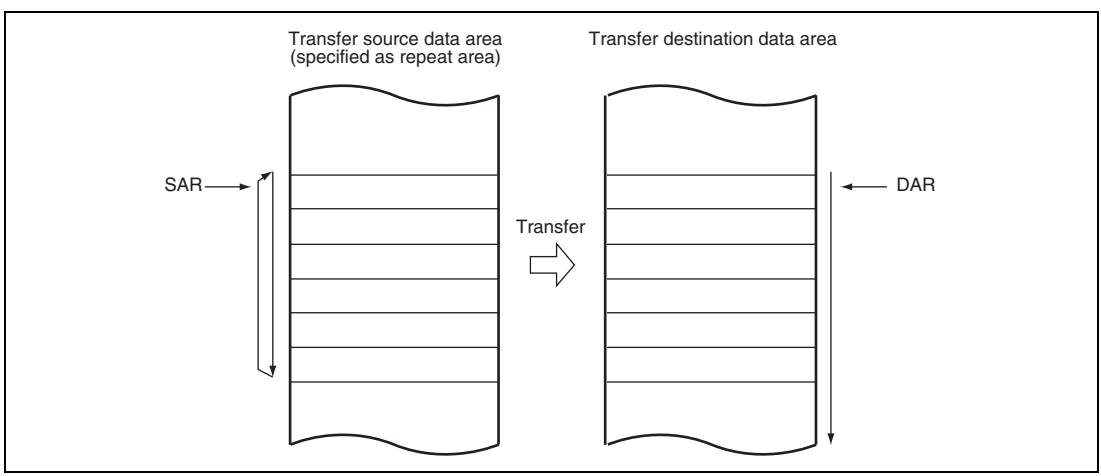



**Figure 8.7 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)**

### 8.5.5 Block Transfer Mode

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

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

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

**Table 8.8 Register Function in Block Transfer Mode**

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

Note: * Transfer information writeback is skipped.

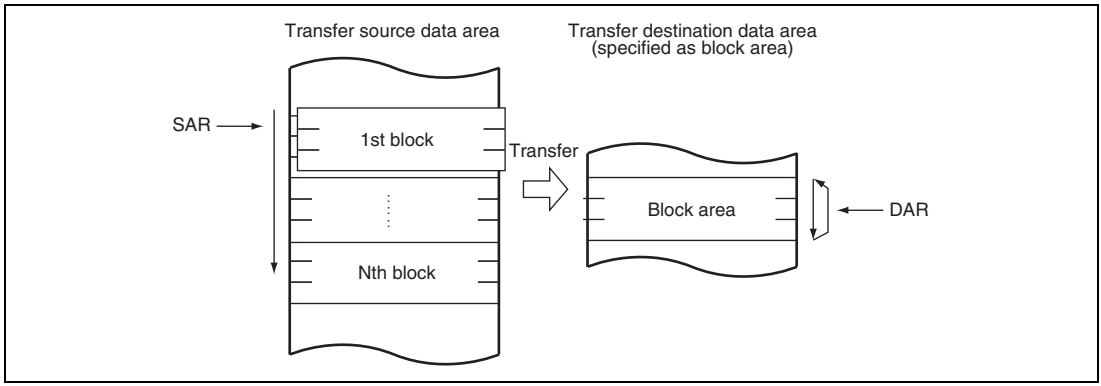



**Figure 8.8 Memory Map in Block Transfer Mode (When Transfer Destination is Specified as Block Area)**

### 8.5.6 Chain Transfer

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

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

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

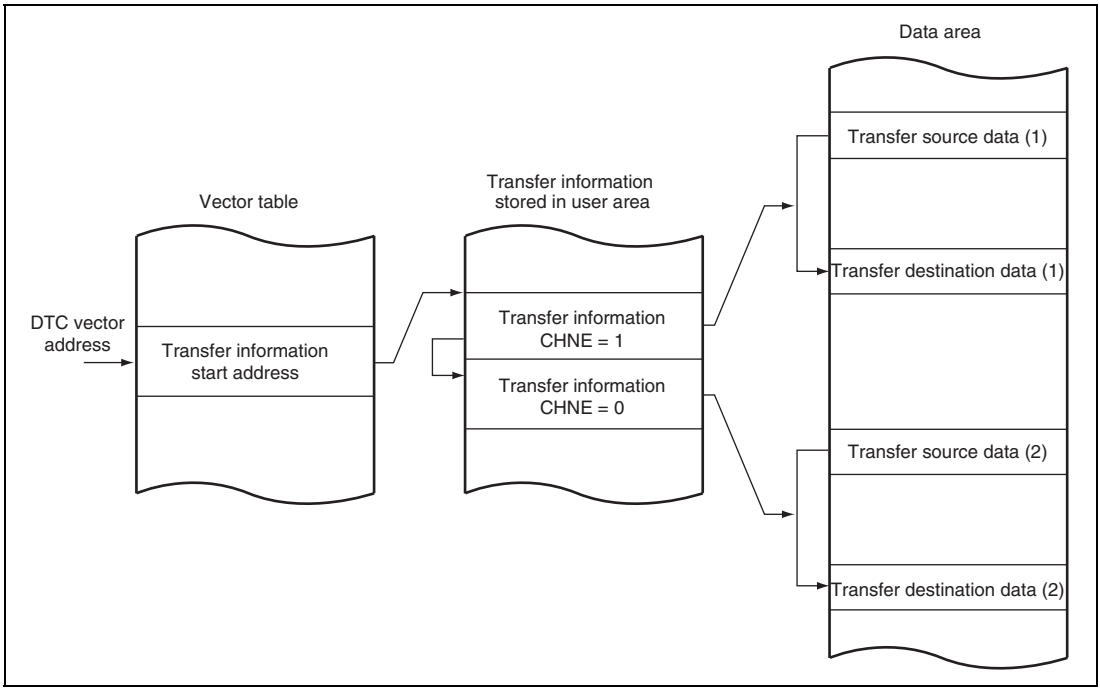
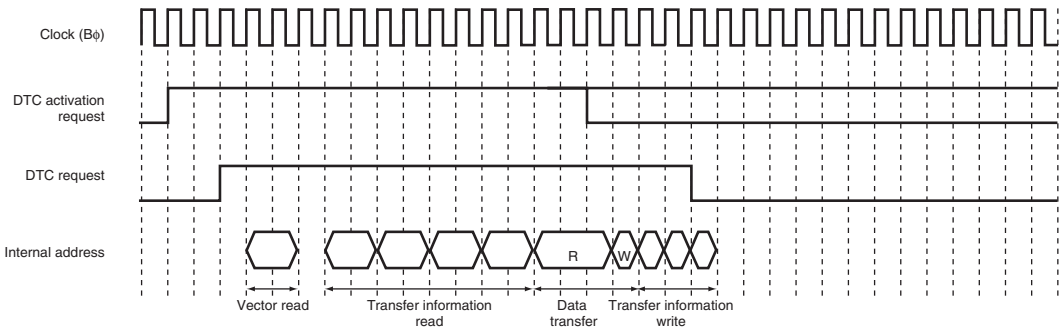



**Figure 8.9 Operation of Chain Transfer**

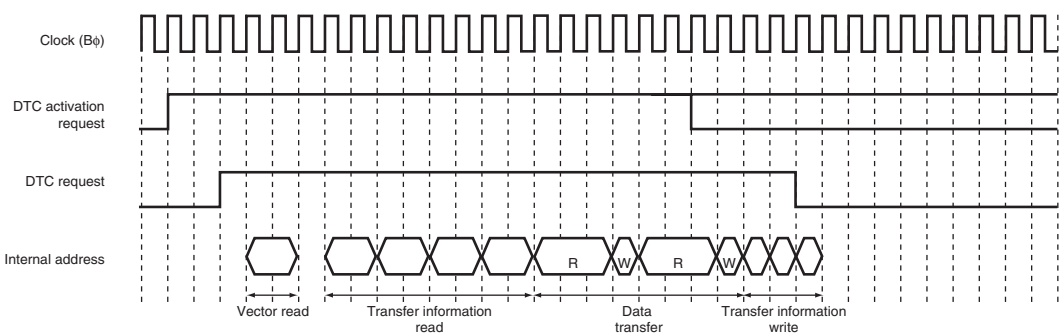
### 8.5.7 Operation Timing

Figures 8.10 to 8.15 show the DTC operation timings.




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

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




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

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

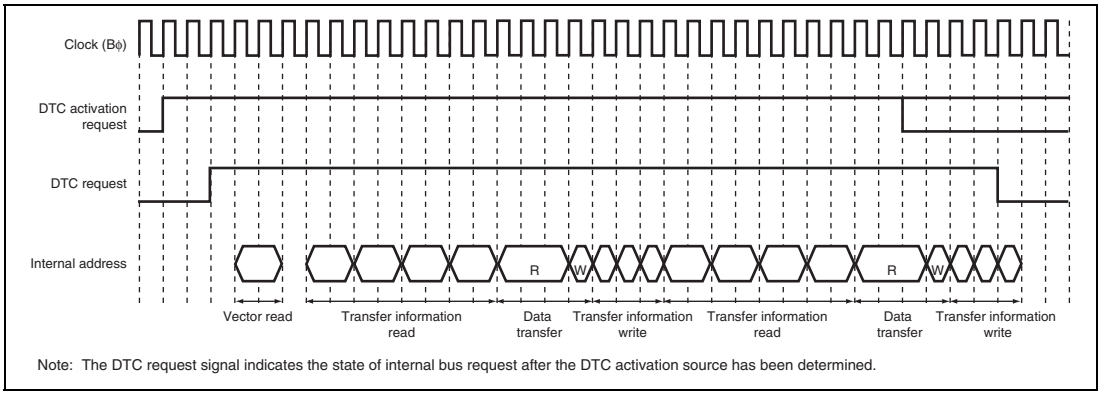



**Figure 8.12 Example of DTC Operation Timing: Chain Transfer (Activated by On-Chip Peripheral Module; Iϕ: Bϕ: Pϕ =1: 1/2: 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)**

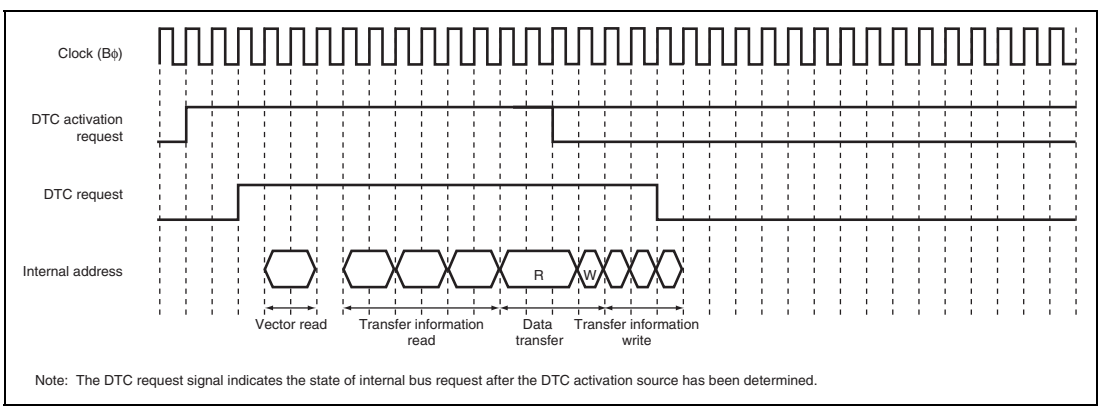



**Figure 8.13 Example of DTC Operation Timing: Normal or Repeat Transfer in Short Address Mode (Activated by On-Chip Peripheral Module; Iϕ: Bϕ: Pϕ =1: 1/2: 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)**

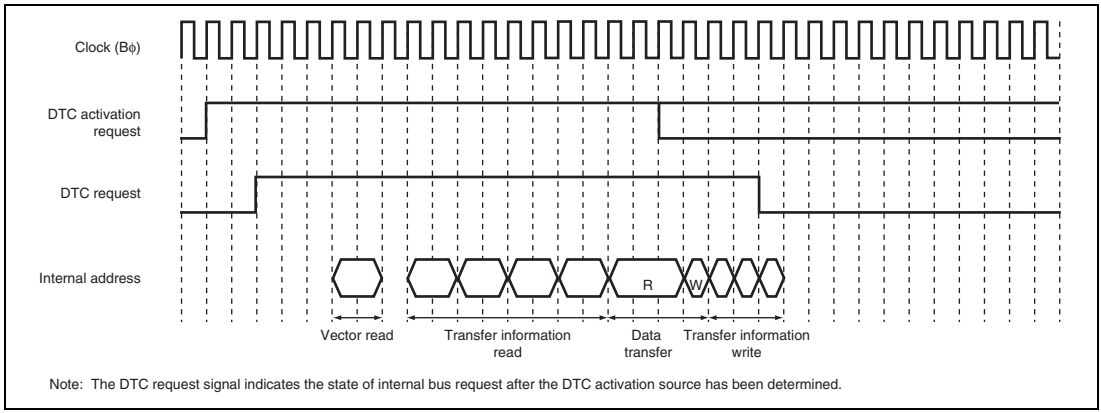



**Figure 8.14 Example of DTC Operation Timing: Normal or Repeat Transfer with DTPR = 1 (Activated by On-Chip Peripheral Module; Iϕ: Bϕ: Pϕ =1: 1/2: 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)**

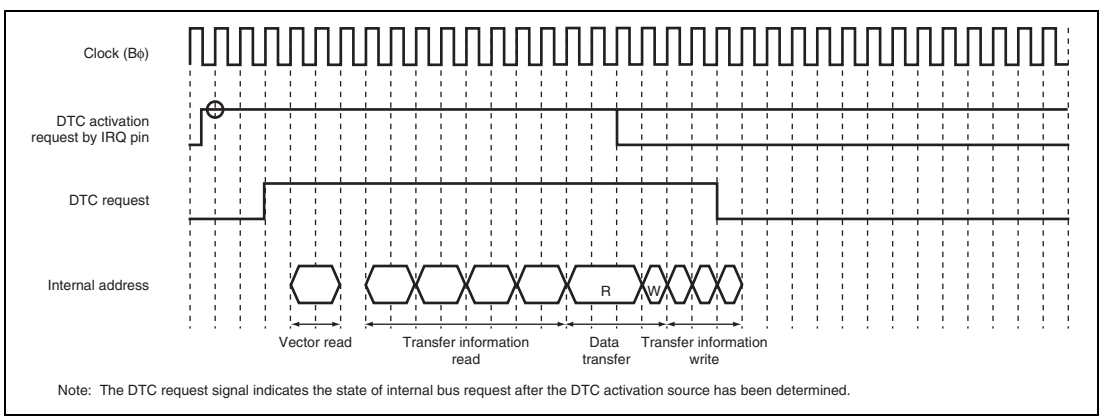



**Figure 8.15 Example of DTC Operation Timing: Normal or Repeat Transfer (Activated by IRQ; Iϕ: Bϕ: Pϕ =1: 1/2: 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)**

### 8.5.8 Number of DTC Execution Cycles

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

**Table 8.9 DTC Execution Status**

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

[Legend]

P: Block size (initial setting of CRAH and CRAL)

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

## Table 8.10 Number of Cycles Required for Each Execution State

| Object to be Accessed | | On-Chip RAM*1/ROM*2 | On-Chip I/O Registers | External Devices*4 | |
|---|---|---|---|---|---|
| Bus width | | 32 bits | 16 bits | 8 bits | 16 bits |
| Access cycles | | 1Bϕ to 3Bϕ*1*2 | 2Pϕ | 2Bϕ | 2Bϕ |
| Execu-tion status | Vector read $S_I$ | 1Bϕ to 3Bϕ*1*2 | — | 9Bϕ | 5Bϕ |
| | Transfer information read $S_J$ | 1Bϕ to 3Bϕ*1 | — | 9Bϕ | 5Bϕ |
| | Transfer information write $S_K$ | 1Bϕ to 3Bϕ*1 | — | 2Bϕ*5 | 2Bϕ*5 |
| | Byte data read $S_L$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 2Pϕ*3 | 3Bϕ | 3Bϕ |
| | Word data read $S_L$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 2Pϕ*3 | 5Bϕ | 3Bϕ |
| | Longword data read $S_L$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 4Pϕ*3 | 9Bϕ | 5Bϕ |
| | Byte data write $S_M$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 2Pϕ*3 | 2Bϕ*5 | 2Bϕ*5 |
| | Word data write $S_M$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 2Pϕ*3 | 2Bϕ*5 | 2Bϕ*5 |
| | Longword data write $S_M$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 4Pϕ*3 | 2Bϕ*5 | 2Bϕ*5 |
| | Internal operation $S_N$ | 1 | | | |

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

| | Read | Write |
|---|---|---|
| Iϕ:Bϕ = 1:1 | 3Bϕ | 3Bϕ |
| Iϕ:Bϕ = 1:1/2 | 2Bϕ | 1Bϕ |
| Iϕ:Bϕ = 1:1/3 | 2Bϕ | 1Bϕ |
| Iϕ:Bϕ = 1:1/4 or less | 1Bϕ | 1Bϕ |

2. Values for on-chip ROM. Number of cycles varies depending on the ratio of Iϕ:Bϕ.and are the same as on-chip RAM. Only vector read is possible.
3. The values in the table are those for the fastest case. Depending on the state of the internal bus, replace 1Bϕ by 1Pϕ in a slow case.
4. Values are different depending on the BSC register setting. The values in the table are the sample for the case with no wait cycles and the WM bit in CSnWCR = 1.
5. Values are different depending on the bus state.

   The number of cycles increases when many external wait cycles are inserted in the case where writing is frequently executed, such as block transfer, and when the external bus is in use because the write buffer cannot be used efficiently in such cases. For details on the write buffer, see section 9.5.7 (2), Access in View of LSI Internal Bus Master.

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

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

### 8.5.9 DTC Bus Release Timing

The DTC requests the bus mastership to the bus arbiter when an activation request occurs. The DTC releases the bus mastership after a vector read, NOP cycle generation after a vector read, transfer information read, a single data transfer, or transfer information writeback. The DTC does not release the bus mastership during transfer information read, single data transfer, or transfer information writeback.
The bus mastership release timing can be specified through the bus function extending register (BSCEHR). For details see section 9.4.4, Bus Function Extending Register (BSCEHR). The difference in bus mastership release timing according to the register setting is summarized in table 8.11. Settings other than settings 1 to 5 are not allowed. The setting must not be changed while the DTC is active.

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

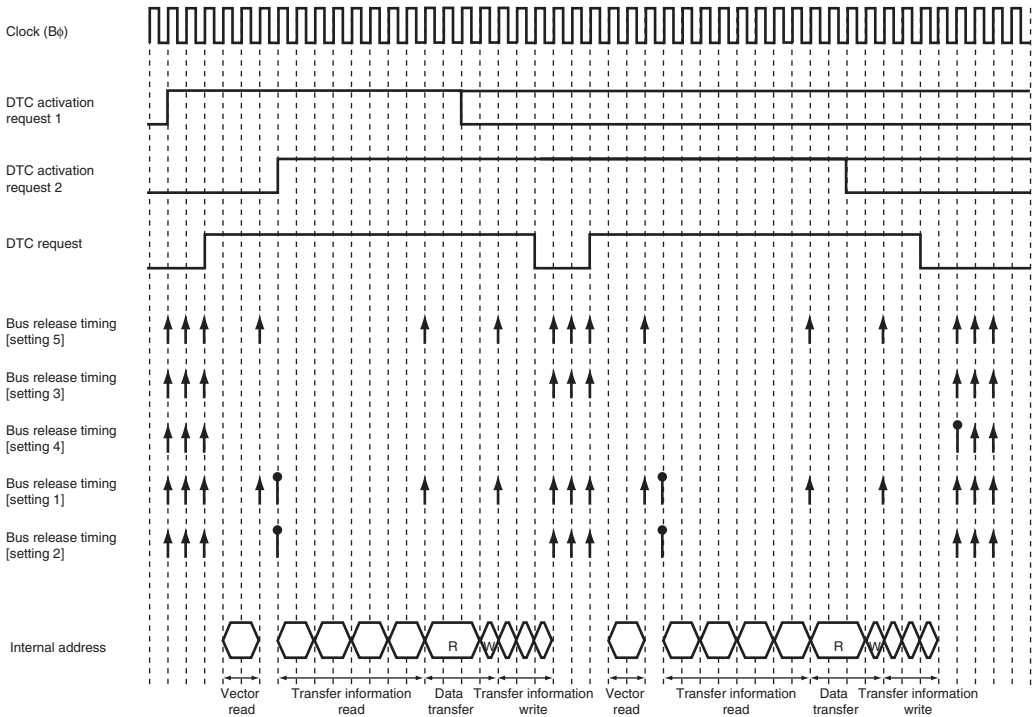
## Table 8.11 DTC Bus Release Timing

| | Bus Function Extending Register (BSCEHR) Setting | | | | | Bus Release Timing (O: Bus is released; x: Bus is not released) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | After write-back of transfer information | |
| Setting | DTLOCK | CSSTP1 | CSSTP2 | CSSTP3 | DTBST | After vector read | NOP cycle generation*1 | After transfer information read | After a single data transfer | Normal transfer | Continuous transfer |
| Setting 1 | 1 | 0 | *3 | 1 | 0 | O | O | O | O | O | O |
| Setting 2 | 0 | 0 | 0 | *3 | 0 | x | O | x | x | O | O |
| Setting 3 | 0 | 1 | *3 | *3 | 0 | x | x | x | x | O | O |
| Setting 4*2 | 0 | 1 | *3 | *3 | 1 | x | x | x | x | O | x |
| Setting 5 | 1 | 1 | *3 | 1 | 0 | O | x | O | O | O | O |

Notes: 1. The bus mastership is only released for the external space access request from the CPU after a vector read.

2. There are following restrictions in setting 4.
   - Clock setting by the frequency control register (FRQCR) must be Iϕ:Bϕ:Pϕ:MIϕ:MPϕ = 8:4:4:4:4, 4:2:2:2:2, or 2:1:1:1:1.
   - Locate vector information in on-chip ROM or on-chip RAM.
   - Locate transfer information in on-chip RAM.
   - Transfer is allowed between on-chip RAM and on-chip peripheral module or between external memory and on-chip peripheral module.

3. Don't care.




[Legend]

↑ : Indicates bus release timing.

• : Bus mastership is only released for the external access request from the CPU.

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

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

### 8.5.10 DTC Activation Priority Order

In the case where multiple DTC activation requests are generated while the DTC is inactive, it is selectable whether the DTC starts transfer in the order of activation request generation or in the order of priority for DTC activation. This selection is made by the setting of the DTPR bit in the bus function extending register (BSCEHR). On the other hand, if multiple activation requests are generated while the DTC is active, transfer is performed according to the priority order for DTC activation. Figure 8.17 shows an example of DTC activation according to the priority.

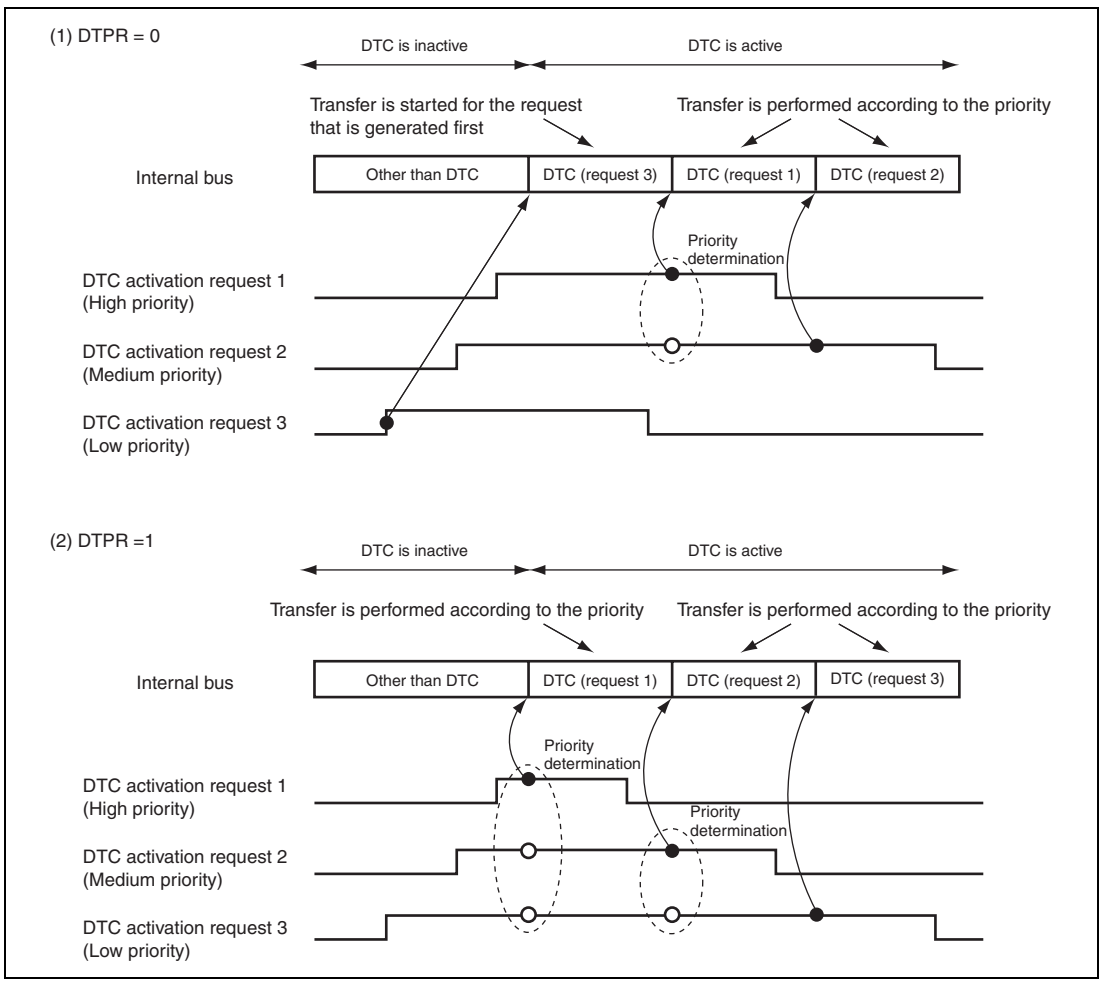
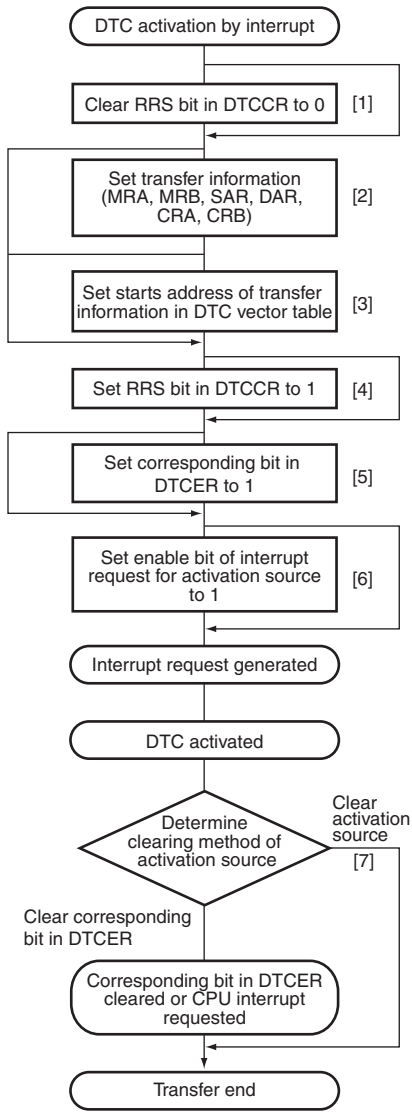



**Figure 8.17 Example of DTC Activation in Accordance with Priority**

## 8.6 DTC Activation by Interrupt

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




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

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

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

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

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

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

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

**Figure 8.18 Activation of DTC by Interrupt**

## 8.7 Examples of Use of the DTC

### 8.7.1 Normal Transfer Mode

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

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

### 8.7.2 Chain Transfer when Counter = 0

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

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

1. For the first transfer, set the normal transfer mode for input data. Set the fixed transfer source address, CRA = H'0000 (65,536 times), CHNE = 1, CHNS = 1, and DISEL = 0.
2. Prepare the upper 8-bit addresses of the start addresses for 65,536-transfer units for the first data transfer in a separate area (in ROM, etc.). For example, if the input buffer is configured at addresses H'200000 to H'21FFFF, prepare H'21 and H'20.

3. For the second transfer, set repeat transfer mode (with the source side as the repeat area) for re-setting the transfer destination address for the first data transfer. Use the upper eight bits of DAR in the first transfer information area as the transfer destination. Set CHNE = DISEL = 0. If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counter to 2.
4. Execute the first data transfer 65536 times by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper eight bits of the transfer destination address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
5. Next, execute the first data transfer the 65536 times specified for the first data transfer by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper eight bits of the transfer destination address for the first data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data transfer, no interrupt request is sent to the CPU.

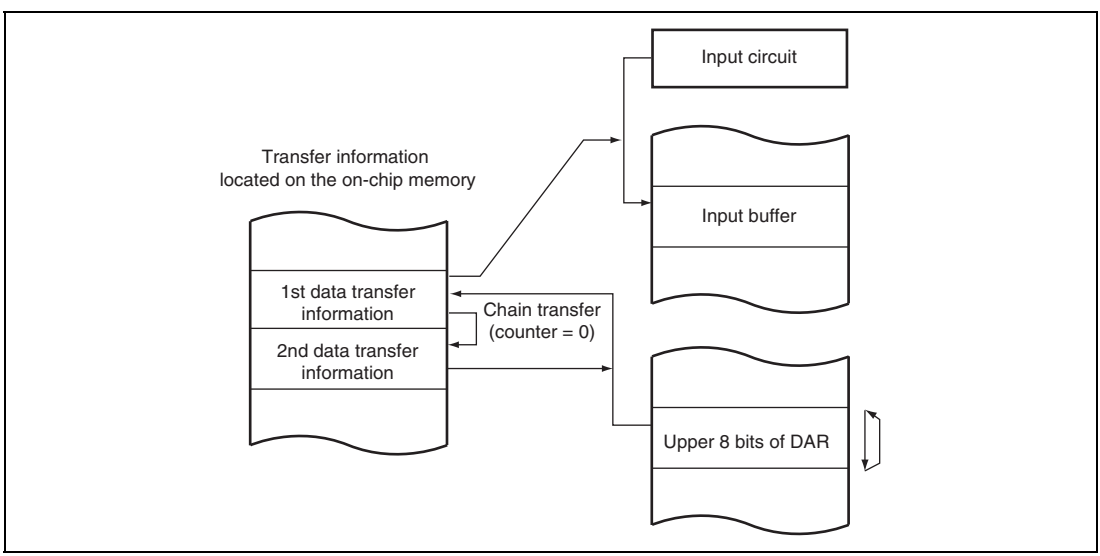



**Figure 8.19 Chain Transfer when Counter = 0**

## 8.8 Interrupt Sources

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or on completion of a single data transfer or a single block data transfer with the DISEL bit set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and priority level control in the interrupt controller. For details, refer to section 6.8, Data Transfer with Interrupt Request Signals.

## 8.9 Usage Notes

### 8.9.1 Module Standby Mode Setting

Operation of the DTC can be disabled or enabled using the standby control register. The initial setting is for operation of the DTC to be disabled. DTC operation is disabled in module standby mode but register access is available. Module standby mode cannot be set while the DTC is activated. Before entering software standby mode or module standby mode, all DTCER registers must be cleared. For details, refer to section 24, Power-Down Modes.

### 8.9.2 On-Chip RAM

Transfer information can be located in on-chip RAM. In this case, the RAME bit in RAMCR must not be cleared to 0.

### 8.9.3 DTCE Bit Setting

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

### 8.9.4 Chain Transfer

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

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

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

### 8.9.6 Access to DTC Registers through DTC

Do not access the DTC registers by using DTC operation.

### 8.9.7 Notes on IRQ Interrupt as DTC Activation Source

- The IRQ interrupt specified as a DTC activation source must not be used to cancel software standby mode.
- The IRQ edge input in software standby mode must not be specified as a DTC activation source.
- When a low level on the IRQ pin is to be detected, if the end of DTC transfer is used to request an interrupt to the CPU (transfer counter = 0 or DISEL = 1), the IRQ signal must be kept low until the CPU accepts the interrupt.

### 8.9.8 Notes on SCI as DTC Activation Sources

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

### 8.9.9 Clearing Interrupt Source Flag

The interrupt source flag set when the DTC transfer is completed should be cleared in the interrupt handler in the same way as for general interrupt source flags. For details, refer to section 6.9, Usage Note.

### 8.9.10 Conflict between NMI Interrupt and DTC Activation

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

It takes 1 × Bcyc + 3 × Pcyc for determining DTC stop by NMI, 2 × Bcyc for determining DTC activation by IRQ, and 1 × Pcyc for determining DTC activation by peripheral modules.

### 8.9.11 Operation When a DTC Activation Request is Cancelled While in Progress

Once the DTC has accepted an activation request, the DTC does not accept the next activation request until the sequence of DTC processing that ends with writeback has been completed.

# Section 9 Bus State Controller (BSC)

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

## 9.1 Features

1. External address space
   - A maximum 1 Mbyte for each of two areas, CS0 and CS1
   - The data bus width is fixed to 8 bits for each address space
   - Controls the insertion of the wait state for each address space.
   - Controls the insertion of the wait state for each read access and write access
   - Can set the independent idling cycle in the continuous access for five cases: read-write (in same space/different space), read-read (in same space/different space), the first cycle is a write access.
2. Normal space interface
   - Supports the interface that can directly connect to the SRAM

Figure 9.1 shows a block diagram of the BSC.

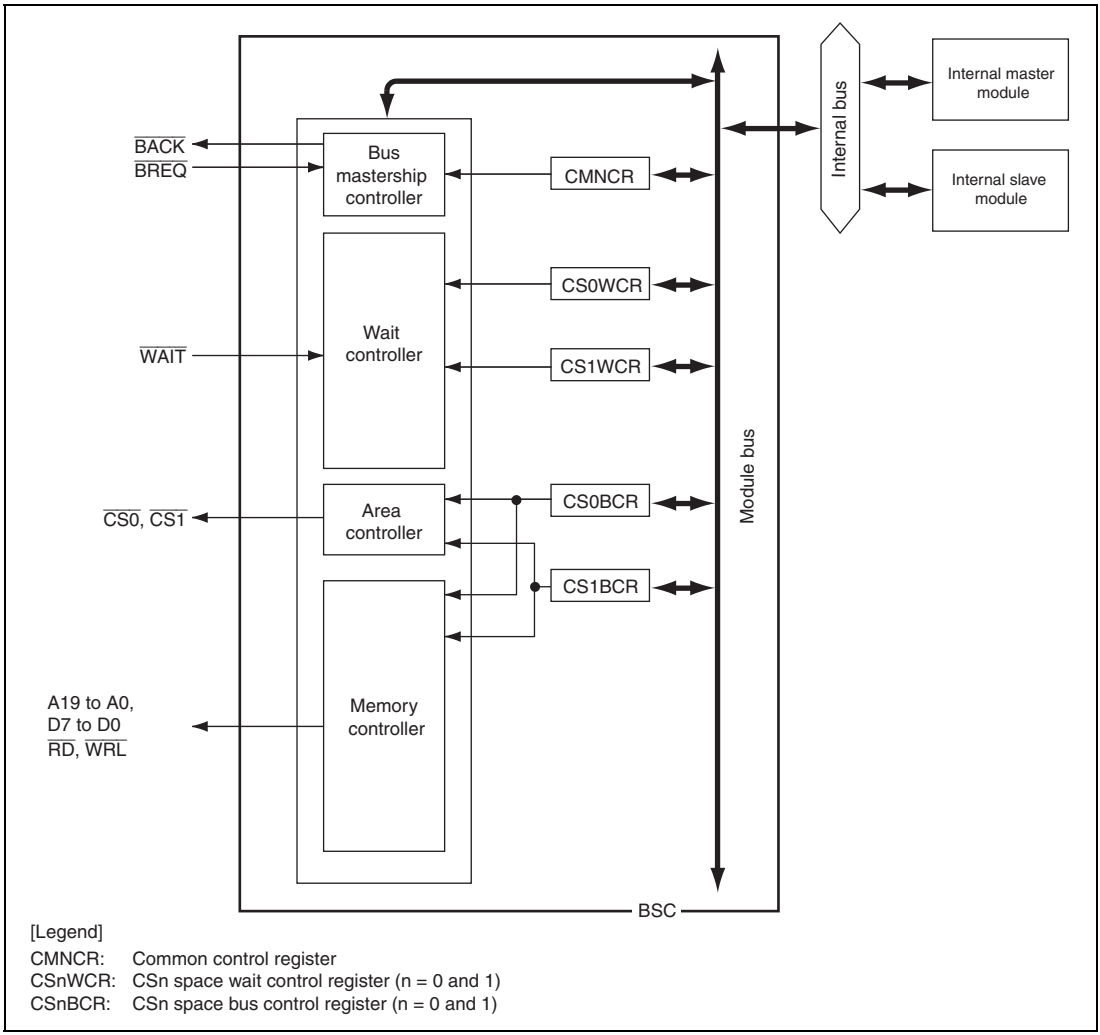



**Figure 9.1 Block Diagram of BSC**

## 9.2 Input/Output Pins

The pin configuration of the BSC is listed in table 9.1.

**Table 9.1 Pin Configuration**

| Name | I/O | Function |
|---|---|---|
| A19 to A0 | Output | Address bus |
| D7 to D0 | I/O | Data bus |
| $\overline{CS0}$ and $\overline{CS1}$ | Output | Chip select |
| $\overline{RD}$ | Output | Read pulse signal (read data output enable signal) |
| $\overline{WRL}$ | Output | Indicates byte write through D7 to D0. |
| $\overline{WAIT}$ | Input | External wait input |
| $\overline{BREQ}$ | Input | Bus request input |
| $\overline{BACK}$ | Output | Bus acknowledge output |

## 9.3 Area Overview

### 9.3.1 Area Division

In the architecture, this LSI has 32-bit address spaces.

As listed in tables 9.2 to 9.7, this LSI can connect two areas to each type of memory, and it outputs chip select signals ($\overline{CS0}$ and $\overline{CS1}$) for each of them. $\overline{CS0}$ is asserted during area 0 access.

### 9.3.2 Address Map

The external address space has a capacity of 2 Mbytes and is used by dividing into two spaces. The memory to be connected and the data bus width are specified in each space. The address map for the entire address space is listed in tables 9.2 to 9.7.

**Table 9.2 Address Map: SH7131/SH7132 (128-Kbyte Flash Memory Version) in Single-Chip Mode**

| Address | Area | Memory Type | Capacity | Bus Width |
|---|---|---|---|---|
| H'00000000 to H'0001FFFF | On-chip ROM | | 128 Kbytes | 32 bits |
| H'00020000 to H'FFFF9FFF | Reserved | | | |
| H'FFFFA000 to H'FFFFBFFF | On-chip RAM | | 8 Kbytes | 32 bits |
| H'FFFFC000 to H'FFFFFFFF | On-chip peripheral modules | | 16 Kbytes | 8 or 16 bits |

Note: Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed. Only the on-chip ROM, on-chip RAM, and on-chip peripheral modules can be accessed; the other areas cannot be accessed.

**Table 9.3 Address Map: SH7131/SH7132/SH7136/SH7137 (256-Kbyte Flash Memory Version) in Single-Chip Mode**

| Address | Area | Memory Type | Capacity | Bus Width |
|---|---|---|---|---|
| H'00000000 to H'0003FFFF | On-chip ROM | | 256 Kbytes | 32 bits |
| H'00040000 to H'FFFF7FFF | Reserved | | | |
| H'FFFF8000 to H'FFFFBFFF | On-chip RAM | | 16 Kbytes | 32 bits |
| H'FFFFC000 to H'FFFFFFFF | On-chip peripheral modules | | 16 Kbytes | 8 or 16 bits |

Note: Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed. Only the on-chip ROM, on-chip RAM, and on-chip peripheral modules can be accessed; the other areas cannot be accessed.

**Table 9.4 Address Map: SH7132 (128-Kbyte Flash Memory Version) in On-Chip ROM-Enabled Mode**

| Address | Area | Memory Type | Capacity | Bus Width |
|---|---|---|---|---|
| H'00000000 to H'0001FFFF | On-chip ROM | | 128 Kbytes | 32 bits |
| H'00020000 to H'01FFFFFF | Reserved | | | |
| H'02000000 to H'020FFFFF | CS0 space | Normal space | 1 Mbyte | 8 bits |
| H'02100000 to H'03FFFFFF | Reserved | | | |
| H'04000000 to H'040FFFFF | CS1 space | Normal space | 1 Mbyte | 8 bits |
| H'04100000 to H'FFFF9FFF | Reserved | | | |
| H'FFFFA000 to H'FFFFBFFF | On-chip RAM | | 8 Kbytes | 32 bits |
| H'FFFFC000 to H'FFFFFFFF | On-chip peripheral modules | | 16 Kbytes | 8 or 16 bits |

Note: Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed. In single-chip mode, only the on-chip ROM, on-chip RAM, and on-chip peripheral modules can be accessed; the other areas cannot be accessed.

**Table 9.5 Address Map: SH7132 (128-Kbyte Flash Memory Version) in On-Chip ROM-Disabled Mode)**

| Address | Area | Memory Type | Capacity | Bus Width |
|---|---|---|---|---|
| H'00000000 to H'000FFFFF | CS0 space | Normal space | 1 Mbyte | 8 bits |
| H'00100000 to H'03FFFFFF | Reserved | | | |
| H'04000000 to H'040FFFFF | CS1 space | Normal space | 1 Mbyte | 8 bits |
| H'04100000 to H'FFFF9FFF | Reserved | | | |
| H'FFFFA000 to H'FFFFBFFF | On-chip RAM | | 8 Kbytes | 32 bits |
| H'FFFFC000 to H'FFFFFFFF | On-chip peripheral modules | | 16 Kbytes | 8 or 16 bits |

Note: Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.

**Table 9.6 Address Map: SH7132/SH7137 (256-Kbyte Flash Memory Version) in On-Chip ROM-Enabled Mode**

| Address | Area | Memory Type | Capacity | Bus Width |
|---|---|---|---|---|
| H'00000000 to H'0003FFFF | On-chip ROM | | 256 Kbytes | 32 bits |
| H'00040000 to H'01FFFFFF | Reserved | | | |
| H'02000000 to H'020FFFFF | CS0 space | Normal space | 1 Mbyte | 8 bits |
| H'02100000 to H'03FFFFFF | Reserved | | | |
| H'04000000 to H'040FFFFF | CS1 space | Normal space | 1 Mbyte | 8 bits |
| H'04100000 to H'FFFF7FFF | Reserved | | | |
| H'FFFF8000 to H'FFFFBFFF | On-chip RAM | | 16 Kbytes | 32 bits |
| H'FFFFC000 to H'FFFFFFFF | On-chip peripheral modules | | 16 Kbytes | 8 or 16 bits |

Note: Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed. In single-chip mode, only the on-chip ROM, on-chip RAM, and on-chip peripheral modules can be accessed; the other areas cannot be accessed.

**Table 9.7 Address Map: SH7132/SH7137 (256-Kbyte Flash Memory Version) in On-Chip ROM-Disabled Mode**

| Address | Area | Memory Type | Capacity | Bus Width |
|---|---|---|---|---|
| H'00000000 to H'000FFFFF | CS0 space | Normal space | 1 Mbyte | 8 bits |
| H'00100000 to H'03FFFFFF | Reserved | | | |
| H'04000000 to H'040FFFFF | CS1 space | Normal space | 1 Mbyte | 8 bits |
| H'04100000 to H'FFFF7FFF | Reserved | | | |
| H'FFFF8000 to H'FFFFBFFF | On-chip RAM | | 16 Kbytes | 32 bits |
| H'FFFFC000 to H'FFFFFFFF | On-chip peripheral modules | | 16 Kbytes | 8 or 16 bits |

Note: Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.

## 9.4 Register Descriptions

The BSC has the following registers. Refer to section 25, List of Registers, for details on the register addresses and register states in each operating mode.

**Table 9.8 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Common control register | CMNCR | R/W | H'00001010 | H'FFFFF000 | 32 |
| CS0 space bus control register | CS0BCR | R/W | H'36DB0600 | H'FFFFF004 | 32 |
| CS1 space bus control register | CS1BCR | R/W | H'36DB0600 | H'FFFFF008 | 32 |
| CS0 space wait control register | CS0WCR | R/W | H'00000500 | H'FFFFF028 | 32 |
| CS1 space wait control register | CS1WCR | R/W | H'00000500 | H'FFFFF02C | 32 |
| Bus function extending register | BSCEHR | R/W | H'0000 | H'FFFFE89A | 8, 16 |

### 9.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | - | - | - | - | HIZMEM | - |
| Initial value: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 13 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 12 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 11 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 4 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | HIZMEM | 0 | R/W | High-Z Memory Control<br>Specifies the pin state in software standby mode for A19 to A0, $\overline{CSn}$, $\overline{WRL}$, and $\overline{RD}$. While the bus is released, these pins are in high-impedance state regardless of this bit setting.<br>0: High impedance in software standby mode<br>1: Driven in software standby mode |
| 0 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

### 9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 and 1)

CSnBCR is a 32-bit readable/writable register that specifies the data bus width of the respective space, and the number of wait cycles between access cycles.

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

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

Note: * When the on-chip ROM is disabled, this bit is 0.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31, 30 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 29, 28 | IWW[1:0] | 11 | R/W | Specification for Idle Cycles between Write-Read/Write-Write Cycles<br>Specify the number of idle cycles to be inserted after access to memory that is connected to the space. The target cycles are write-read cycles and write-write cycles.<br>00: No idle cycle inserted<br>01: 1 idle cycle inserted<br>10: 2 idle cycles inserted<br>11: 4 idle cycles inserted |
| 27 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 26, 25 | IWRWD[1:0] | 11 | R/W | Specification for Idle Cycles between Read-Write Cycles in Different Spaces<br>Specify the number of idle cycles to be inserted after access to memory that is connected to the space. The target cycles are continuous read-write cycles in different spaces.<br>00: No idle cycle inserted<br>01: 1 idle cycle inserted<br>10: 2 idle cycles inserted<br>11: 4 idle cycles inserted |
| 24 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 23, 22 | IWRWS[1:0] | 11 | R/W | Specification for Idle Cycles between Read-Write Cycles in the Same Space<br>Specify the number of idle cycles to be inserted after access to memory that is connected to the space. The target cycles are continuous read-write cycles in the same space.<br>00: No idle cycle inserted<br>01: 1 idle cycle inserted<br>10: 2 idle cycles inserted<br>11: 4 idle cycles inserted |
| 21 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 20, 19 | IWRRD[1:0] | 11 | R/W | Specification for Idle Cycles between Read-Read Cycles in Different Spaces<br>Specify the number of idle cycles to be inserted after access to memory that is connected to the space. The target cycles are continuous read-read cycles in different spaces.<br>00: No idle cycle inserted<br>01: 1 idle cycle inserted<br>10: 2 idle cycles inserted<br>11: 4 idle cycles inserted |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 18 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 17, 16 | IWRRS[1:0] | 11 | R/W | Specification for Idle Cycles between Read-Read Cycles in the Same Space<br>Specify the number of idle cycles to be inserted after access to memory that is connected to the space. The target cycles are continuous read-read cycles in the same space.<br>00: No idle cycle inserted<br>01: 1 idle cycle inserted<br>10: 2 idle cycles inserted<br>11: 4 idle cycles inserted |
| 15 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 10, 9 | BSZ[1:0] | 01/11* | R/W | Data Bus Size Specification<br>Specify the data bus size of the space. When the on-chip ROM is enabled, write B'01 to specify the data bus width as 8-bit before accessing the CSn space.<br>Note: When the on-chip ROM is disabled, the data bus width of area 0 is 8 bits regardless of the BSZ[1:0] bit setting in CS0BCR. |
| 8 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

Note: * B'01 when the on-chip ROM is disabled.

### 9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 and 1)

CSnWCR specifies various wait cycles for memory accesses. Specify CSnWCR before accessing the target area. CSnWCR should be modified only after CSnBCR setting is completed.

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 19 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 18 to 16 | WW[2:0] | 000 | R/W | Number of Wait Cycles in Write Access<br>Specify the number of cycles required for write access.<br>000: The same cycles as WR3 to WR0 settings (read access wait)<br>001: 0 cycles<br>010: 1 cycle<br>011: 2 cycles<br>100: 3 cycles<br>101: 4 cycles<br>110: 5 cycles<br>111: 6 cycles |
| 15 to 13 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 12, 11 | SW[1:0] | 00 | R/W | Number of Delay Cycles from Address and $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$ and $\overline{\text{WRL}}$ Assertion<br>Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WRL}}$ assertion.<br>00: 0.5 cycle<br>01: 1.5 cycles<br>10: 2.5 cycles<br>11: 3.5 cycles |
| 10 to 7 | WR[3:0] | 1010 | R/W | Number of Read Access Wait Cycles<br>Specify the number of wait cycles required for read access.<br>0000: 0 cycles<br>0001: 1 cycle<br>0010: 2 cycles<br>0011: 3 cycles<br>0100: 4 cycles<br>0101: 5 cycles<br>0110: 6 cycles<br>0111: 8 cycles<br>1000: 10 cycles<br>1001: 12 cycles<br>1010: 14 cycles<br>1011: 18 cycles<br>1100: 24 cycles<br>1101: Reserved (setting prohibited)<br>1110: Reserved (setting prohibited)<br>1111: Reserved (setting prohibited) |
| 6 | WM | 0 | R/W | External Wait Mask Specification<br>Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0.<br>0: External wait input is valid<br>1: External wait input is ignored |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1, 0 | HW[1:0] | 00 | R/W | Delay Cycles from $\overline{RD}$ and $\overline{WRL}$ Negation to Address and $\overline{CSn}$ Negation<br>Specify the number of delay cycles from $\overline{RD}$ and $\overline{WRL}$ negation to address and $\overline{CSn}$ negation.<br>00: 0.5 cycle<br>01: 1.5 cycles<br>10: 2.5 cycles<br>11: 3.5 cycles |

### 9.4.4 Bus Function Extending Register (BSCEHR)

BSCEHR is a 16-bit register that specifies the timing of bus release by the DTC. It also specifies the application of priority in transfer operations and enables or disables the functions that have the effect of decreasing numbers of cycles over which the DTC is active. The differences in DTC operation made by the combinations of the DTLOCK, CSSTP1, and DTBST bits settings are described in section 8.5.9, DTC Bus Release Timing.

Setting the CSSTP2 bit can improve the transfer performance of the DTC transfer when the DTLOCK bit is 0. Furthermore, setting the CSSTP3 bit selects whether or not access to the external space by the CPU takes priority over DTC transfer.

The DTC short address mode is implemented by setting the DTSA bit. For details of the short address mode, see section 8.4, Location of Transfer Information and DTC Vector Table.

A DTC activation priority order can be set up for the DTC activation sources. The DTPR bit selects whether or not this priority order is valid or invalid when multiple sources issue activation requests before DTC activation. Do not modify this register while the DTC is active.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DTLOCK | CSSTP1 | - | CSSTP2 | DTBST | DTSA | CSSTP3 | DTPR | - | - | - | - | - | - | - | - |
| Initial 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/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | DTLOCK | 0 | R/W | DTC Lock Enable<br>Specifies the timing of bus release by the DTC.<br>0: The DTC releases the bus on generation of the NOP cycle that follows vector read or write-back of transfer information.<br>1: The DTC releases the bus after vector read, on generation of the NOP cycle that follows vector read, after transfer information read, after a round of data transfer, or after write-back of transfer information. |
| 14 | CSSTP1 | 0 | R/W | Select Bus Release on NOP Cycle Generation by DTC<br>Specifies whether or not the bus is released in response to requests from the CPU for external space access on generation of the NOP cycle that follows reading of the vector address.<br>If, however, the CSSTP2 bit is 1, bus mastership is retained until all transfer is complete, regardless of the setting of this bit.<br>0: The bus is released on generation of the NOP cycle by the DTC.<br>1: The bus is not released on generation of the NOP cycle by the DTC. |
| 13 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 12 | CSSTP2 | 0 | R/W | Select Bus Release during DTC Transfer<br>This setting applies to DTC transfer when the DTLOCK bit is 0. The value specifies whether the bus mastership is or is not to be released after each round of transfer in response to a request from the CPU for access to the external space.<br>0: When the DTLOCK and CSSTP1 bits are 0, the bus is released on generation of the NOP cycle after reading of the vector address. When the DTLOCK bit is 0 and the CSSTP1 bit is 1, the bus is released after each round of data transfer.<br>1: Only release the bus mastership after all data transfer is complete. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 11 | DTBST | 0 | R/W | DTC Burst Enable<br>Selects whether or not the DTC retains the bus mastership and remains continuously active until all transfer operations are complete when multiple DTC activation requests have been generated.<br>0: Release the bus on the completion of transfer for each individual DTC activation source.<br>1: Keep the DTC continuously active, i.e. only release the bus on completion of processing for all DTC activation sources.<br>Notes: When this bit is set to 1, the following restrictions apply.<br>1. Clock setting with the frequency control register (FRQCR) must be Iϕ: Bϕ: Pϕ: MIϕ: MPϕ: = 8: 4: 4: 4: 4, 4: 2: 2: 2: 2, or 2: 1: 1: 1: 1<br>2. The vector information must be in on-chip ROM or on-chip RAM.<br>3. The transfer information must be in on-chip RAM.<br>4. Transfer must be between the on-chip RAM and an on-chip peripheral module or between external memory and an on-chip peripheral module. |
| 10 | DTSA | 0 | R/W | DTC Short Address Mode<br>In this mode, the information that specifies a DTC transfer takes up only 3 longwords.<br>0: Transfer information is read out as 4 longwords. The transfer information is arranged as shown in figure 8.2 (normal address mode).<br>1: Transfer information is read out as 3 longwords. The transfer information is arranged as shown in figure 8.2 (short address mode).<br>Note: Transfer in short address mode is only available between on-chip peripheral modules and on-chip RAM, because the higher-order 8 bits of the SAR and DAR are considered to be all 1. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 9 | CSSTP3 | 0 | R/W | Select Priority for External Memory Access by CPU<br>Specifies whether or not access to the external space by the CPU takes priority over DTC transfer.<br>0: DTC transfer has priority.<br>1: External space access from the CPU has priority.<br>Note: When this bit is 0, and access to internal I/O from the CPU is immediately followed by access to external space from the CPU, a NOP 1Bϕ in duration is inserted between the two access cycles. |
| 8 | DTPR | 0 | R/W | Application of Priority in DTC Activation<br>When multiple DTC activation requests are generated before the DTC is activated, specify whether transfer starts from the first request to have been generated or is in accord with the priority order for DTC activation requests.<br>However, when multiple DTC activation requests have been issued while the DTC is active, the next transfer to be triggered will be that with the highest DTC activation priority.<br>0: Start transfer in response to the first request to have been generated.<br>1: Start transfer in accord with DTC activation request priority.<br>Notes: When this bit is set to 1, the following restrictions apply.<br>1. The vector information must be in on-chip ROM or on-chip RAM.<br>2. The transfer information must be in on-chip RAM.<br>3. Skipping of transfer information reading is always disabled. |
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

## 9.5 Operation

### 9.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian, in which the 0 address is the most significant byte (MSB) in the byte data.

The data bus width is 8 bits. Data alignment is performed in accordance with the data bus width of the respective device. This also means that when longword data is read from a byte-width device, the read operation must be done four times. In this LSI, data alignment and conversion of data length are performed automatically between the respective interfaces.

Table 9.9 shows the relationship between device data width and access unit.

**Table 9.9 8-Bit External Device Access and Data Alignment**

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

### 9.5.2 Normal Space Interface

**Basic Timing:** For access to a normal space, this LSI uses strobe signal output in consideration of the fact that mainly SRAM without a byte selection will be directly connected. Figure 9.2 shows the basic timings of normal space access. A no-wait normal access is completed in two cycles.

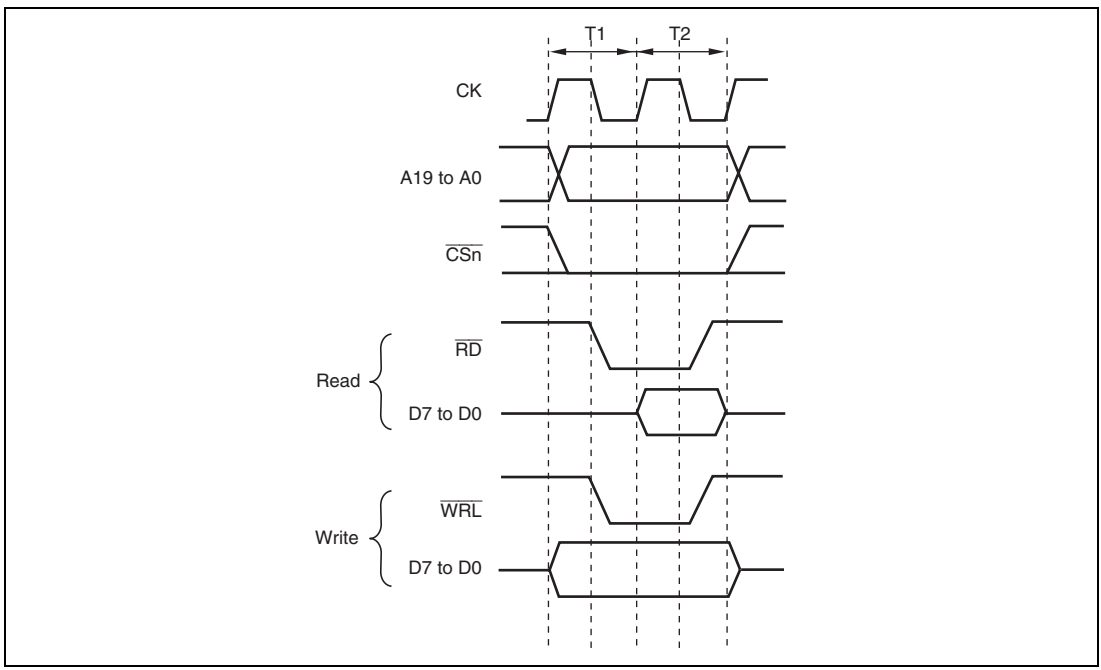



**Figure 9.2 Normal Space Basic Access Timing (Access Wait 0)**

It is necessary to control of outputing the data that has been read using $\overline{RD}$ when a buffer is established in the data bus.

Figures 9.3 and 9.4 show the basic timings of continuous accesses to normal space. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted to evaluate the external wait (figure 9.3). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 9.4).

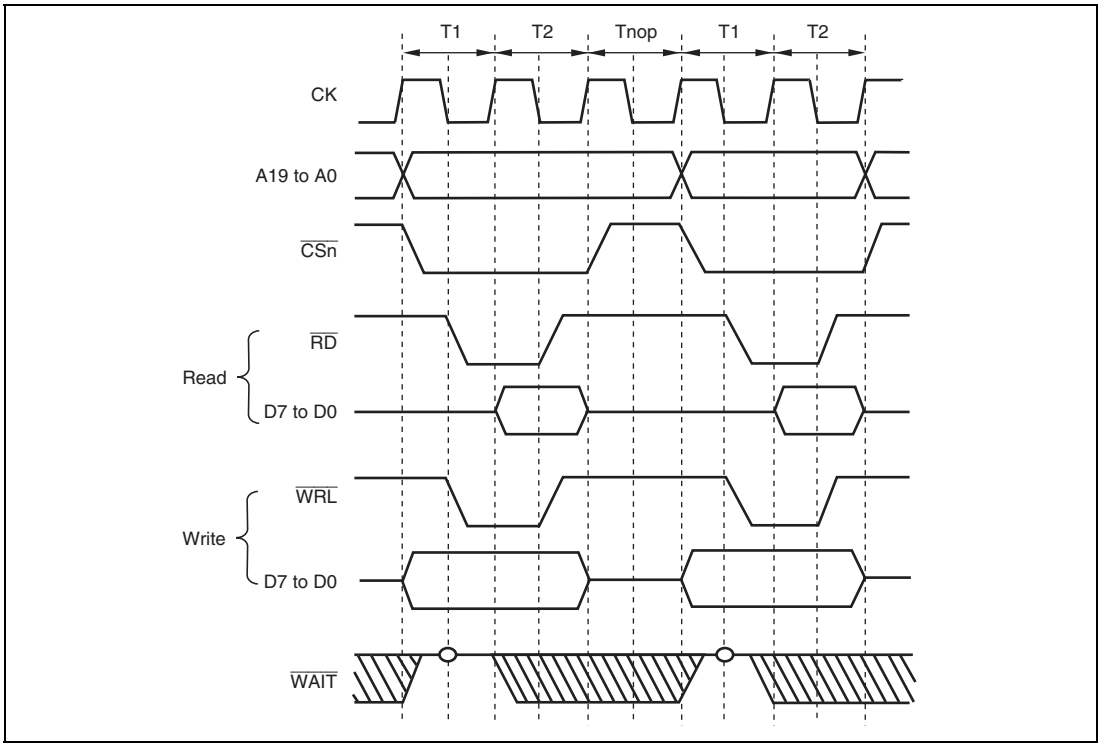



**Figure 9.3 Continuous Access for Normal Space 1**
**Bus Width = 8 Bits, Word Access, WM Bit in CSnWCR = 0**
**(Access Wait = 0, Cycle Wait = 0)**

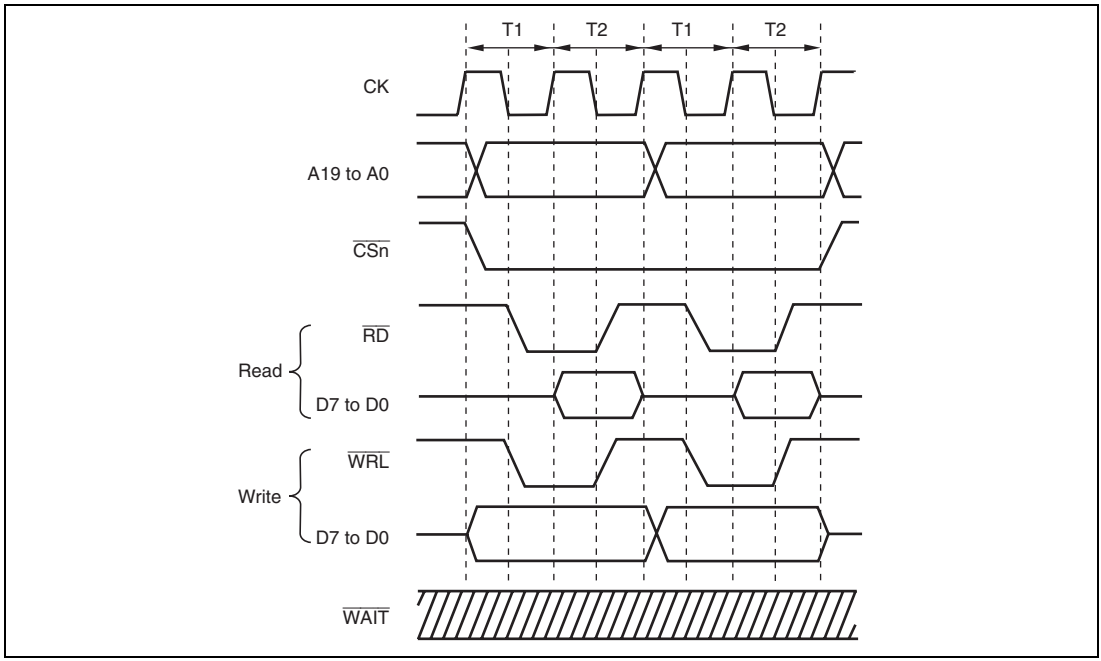



**Figure 9.4 Continuous Access for Normal Space 2**
**Bus Width = 8 Bits, Word Access, WM Bit in CSnWCR = 1**
**(Access Wait = 0, Cycle Wait = 0)**

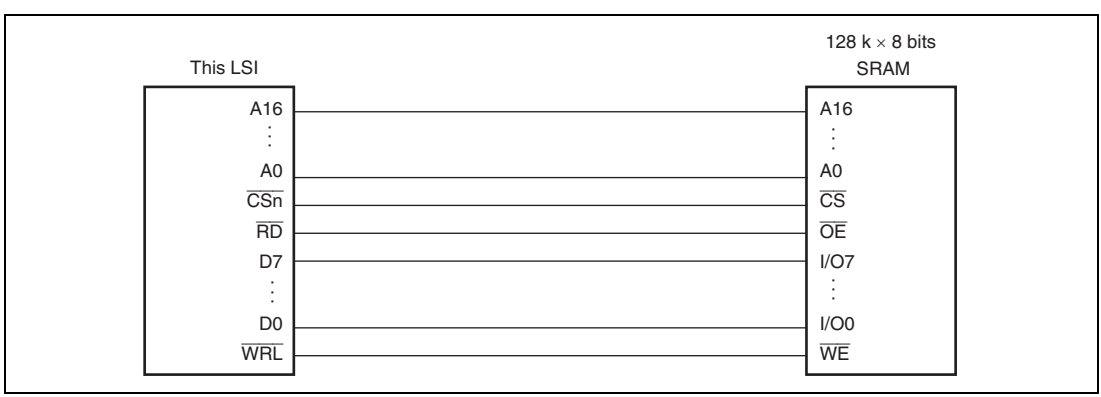



**Figure 9.5 Example of 8-Bit Data-Width SRAM Connection**

### 9.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible to insert wait cycles independently in read access and in write access. The specified number of Tw cycles is inserted as wait cycles in a normal space access shown in figure 9.6.

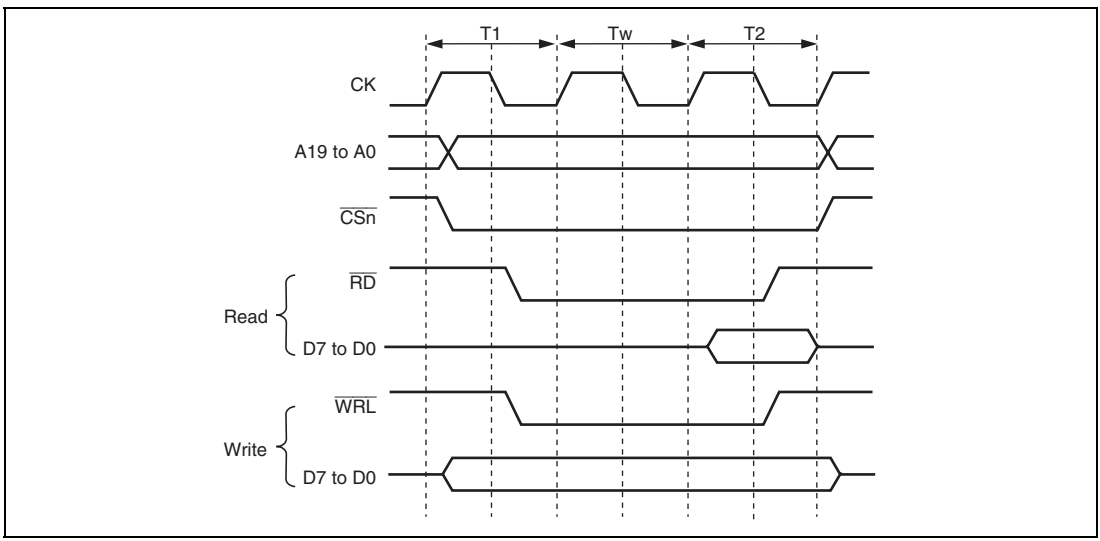



**Figure 9.6 Wait Timing for Normal Space Access (Software Wait Only)**

When the WM bit in CSnWCR is cleared to 0, the external wait input $\overline{\text{WAIT}}$ signal is also sampled. $\overline{\text{WAIT}}$ pin sampling is shown in figure 9.7. A 2-cycle wait is specified as a software wait. The $\overline{\text{WAIT}}$ signal is sampled at the falling edge of CK at the transition from the T1 or Tw cycle to the T2 cycle.

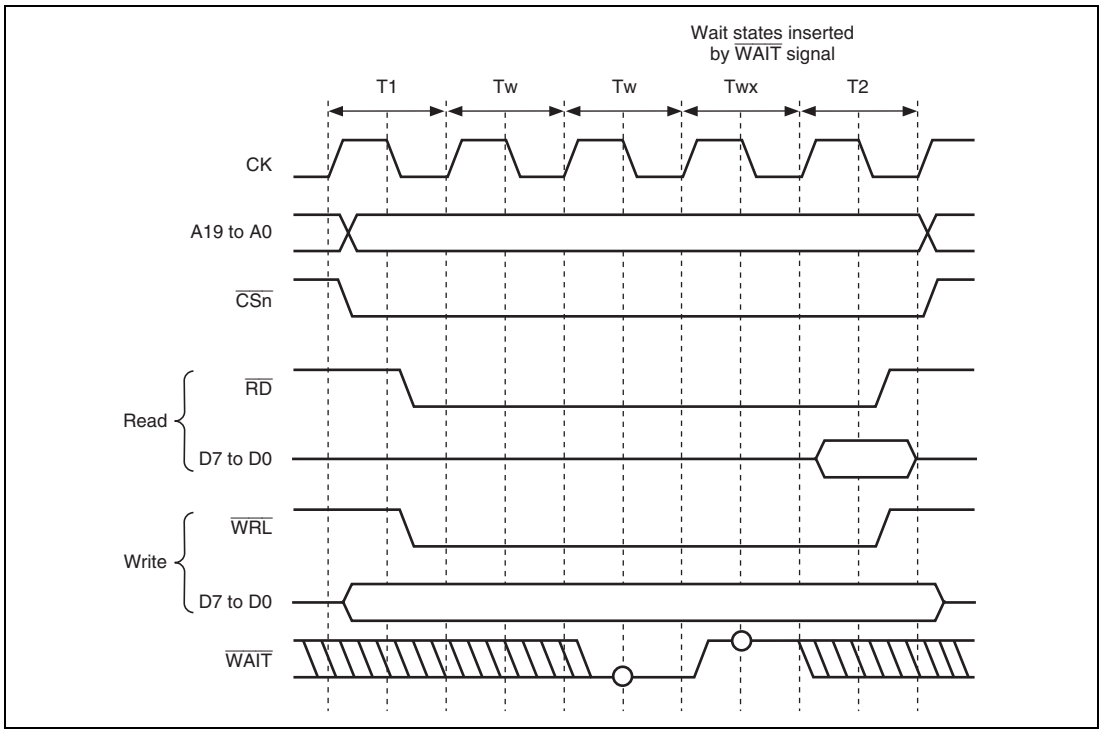



**Figure 9.7 Wait State Timing for Normal Space Access (Wait State Insertion Using $\overline{\text{WAIT}}$ Signal)**

### 9.5.4 $\overline{\text{CSn}}$ Assert Period Extension

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$, $\overline{\text{WRL}}$ assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$, $\overline{\text{WRL}}$ negation to $\overline{\text{CSn}}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 9.8 shows an example. A Th cycle and a Tf cycle are added before and after an ordinary cycle, respectively. In these cycles, $\overline{\text{RD}}$ and $\overline{\text{WRL}}$ are not asserted, while other signals are asserted. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.

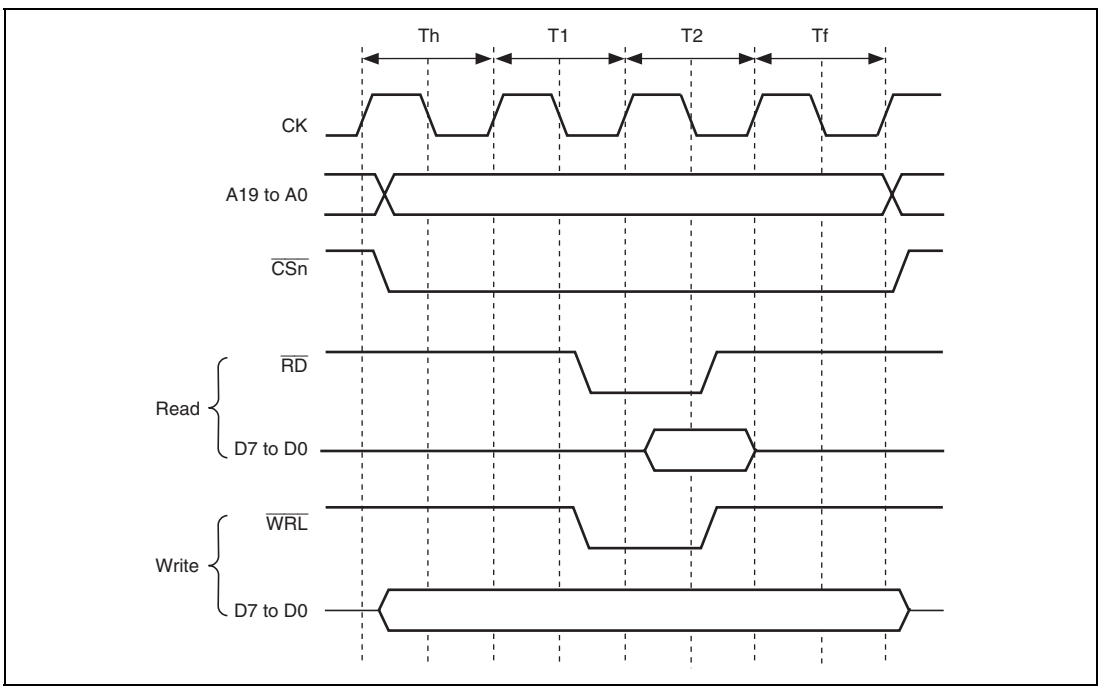



**Figure 9.8 $\overline{\text{CSn}}$ Assert Period Extension**

### 9.5.5 Wait between Access Cycles

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

The number of wait cycles between access cycles can be set by bits IWW[1:0], IWRWD[1:0], IWRWS[1:0], IWRRD[1:0], and IWRRS[1:0] in CSnBCR. The conditions for setting the wait cycles between access cycles (idle cycles) are shown below.

1. Continuous accesses are write-read or write-write
2. Continuous accesses are read-write for different spaces
3. Continuous accesses are read-write for the same space
4. Continuous accesses are read-read for different spaces
5. Continuous accesses are read-read for the same space

Besides the wait cycles between access cycles (idle cycles) described above, idle cycles must be inserted to reserve the minimum pulse width for a multiplexed pin ($\overline{\text{WRL}}$), and an interface with an internal bus.

6. Idle cycle of the external bus for the interface with the internal bus
   A. Insert one idle cycle immediately before a write access cycle after an external bus idle cycle or a read cycle.
   B. Insert one idle cycle to transfer the read data to the internal bus when a read cycle of the external bus terminates.

      Insert two to three idle cycles including the idle cycle in A. for the write cycle immediately after a read cycle.

Tables 9.10 and 9.11 list the minimum number of idle cycles to be inserted. The CSnBCR Idle Setting column in the tables describes the number of idle cycles to be set for IWW, IWRWD, IWRWS, IWRRD, and IWRRS.

## Table 9.10 Minimum Number of Idle Cycles between CPU Access Cycles in Normal Space Interface

| BSC Register Setting | | When Access Size Exceeds Bus Width | | | | | |
|---|---|---|---|---|---|---|---|
| **CSnWCR.WM Setting** | **CSnBCR Idle Setting** | **Contin-uous Read*1** | **Contin-uous Write*1** | **Read to Read*2** | **Write to Write*2** | **Read to Write*2** | **Write to Read*2** |
| 1 | 0 | 0/0/0/0 | 0/0/0/0 | 1/1/1/1 | 0/0/0/0 | 3/3/3/4 | 0/0/0/0 |
| 0 | 0 | 1/1/1/1 | 1/1/1/1 | 1/1/1/1 | 1/1/1/1 | 3/3/3/4 | 1/1/1/1 |
| 1 | 1 | 1/1/1/1 | 1/1/1/1 | 1/1/1/1 | 1/1/1/1 | 3/3/3/4 | 1/1/1/1 |
| 0 | 1 | 1/1/1/1 | 1/1/1/1 | 1/1/1/1 | 1/1/1/1 | 3/3/3/4 | 1/1/1/1 |
| 1 | 2 | 2/2/2/2 | 2/2/2/2 | 2/2/2/2 | 2/2/2/2 | 3/3/3/4 | 2/2/2/2 |
| 0 | 2 | 2/2/2/2 | 2/2/2/2 | 2/2/2/2 | 2/2/2/2 | 3/3/3/4 | 2/2/2/2 |
| 1 | 4 | 4/4/4/4 | 4/4/4/4 | 4/4/4/4 | 4/4/4/4 | 4/4/4/4 | 4/4/4/4 |
| 0 | 4 | 4/4/4/4 | 4/4/4/4 | 4/4/4/4 | 4/4/4/4 | 4/4/4/4 | 4/4/4/4 |

Notes: The minimum numbers of idle cycles are described sequentially for Iϕ:Bϕ = 4:1, 3:1, 2:1, and 1:1.

1. Minimum number of idle cycles between the byte access to address 0 and the byte access to address 1 in the 16-bit access with an 8-bit bus width, and minimum number of idle cycles between the byte accesses to address 0, to address 1, to address 2, and to address 3 in the 32-bit access with an 8-bit bus width.
2. Other than the above cases

**Table 9.11 Minimum Number of Idle Cycles between Access Cycles during DTC Transfer for the Normal Space Interface**

| BSC Register Setting | | When Access Size Exceeds Bus Width | | | |
|---|---|---|---|---|---|
| **CSnWCR. WM Setting** | **CSnBCR Idle Setting** | **Continuous Read*1** | **Read to Write*2** | **Continuous Write*1** | **Write to Read*2** |
| 1 | 0 | 0 | 2 | 0 | 0 |
| 0 | 0 | 1 | 2 | 1 | 1 |
| 1 | 1 | 1 | 2 | 1 | 1 |
| 0 | 1 | 1 | 2 | 1 | 1 |
| 1 | 2 | 2 | 2 | 2 | 2 |
| 0 | 2 | 2 | 2 | 2 | 2 |
| 1 | 4 | 4 | 4 | 4 | 4 |
| 0 | 4 | 4 | 4 | 4 | 4 |

Notes: DTC is operated by Bϕ. The minimum number of idle cycles is not affected by changing a clock ratio.

1. Minimum number of idle cycles between the byte access to address 0 and the byte access to address 1 in the 16-bit access with an 8-bit bus width, and minimum number of idle cycles between the byte accesses to address 0, to address 1, to address 2, and to address 3 in the 32-bit access with an 8-bit bus width.
2. Other than the above cases.

### 9.5.6 Bus Arbitration

This LSI owns the bus mastership in normal state and releases the bus only when receiving a bus request from an external device. This LSI has two bus masters: CPU and DTC. The bus mastership is given to these bus masters in accordance with the following priority.

Request for bus mastership by external device ($\overline{\text{BREQ}}$) > CPU > DTC > CPU

However, when DTC is requesting the bus mastership, the CPU does not obtain the bus mastership continuously.

When the CSSTP2 bit is 1 in the bus function extending register (BSCHER), the external space access request from the CPU has lower priority than the DTC transfer request with DTLOCK = 0 in the bus function extending register (BSCHER).

In addition, because the write buffer operates as described in section 9.5.7 (2), Access in View of LSI Internal Bus Master, arbitration between the CPU and DTC is different depending on whether the external space access by the CPU is a write or read access. Figure 9.9 shows the bus arbitration when a DTC activation request is generated while an external space is accessed by the CPU.

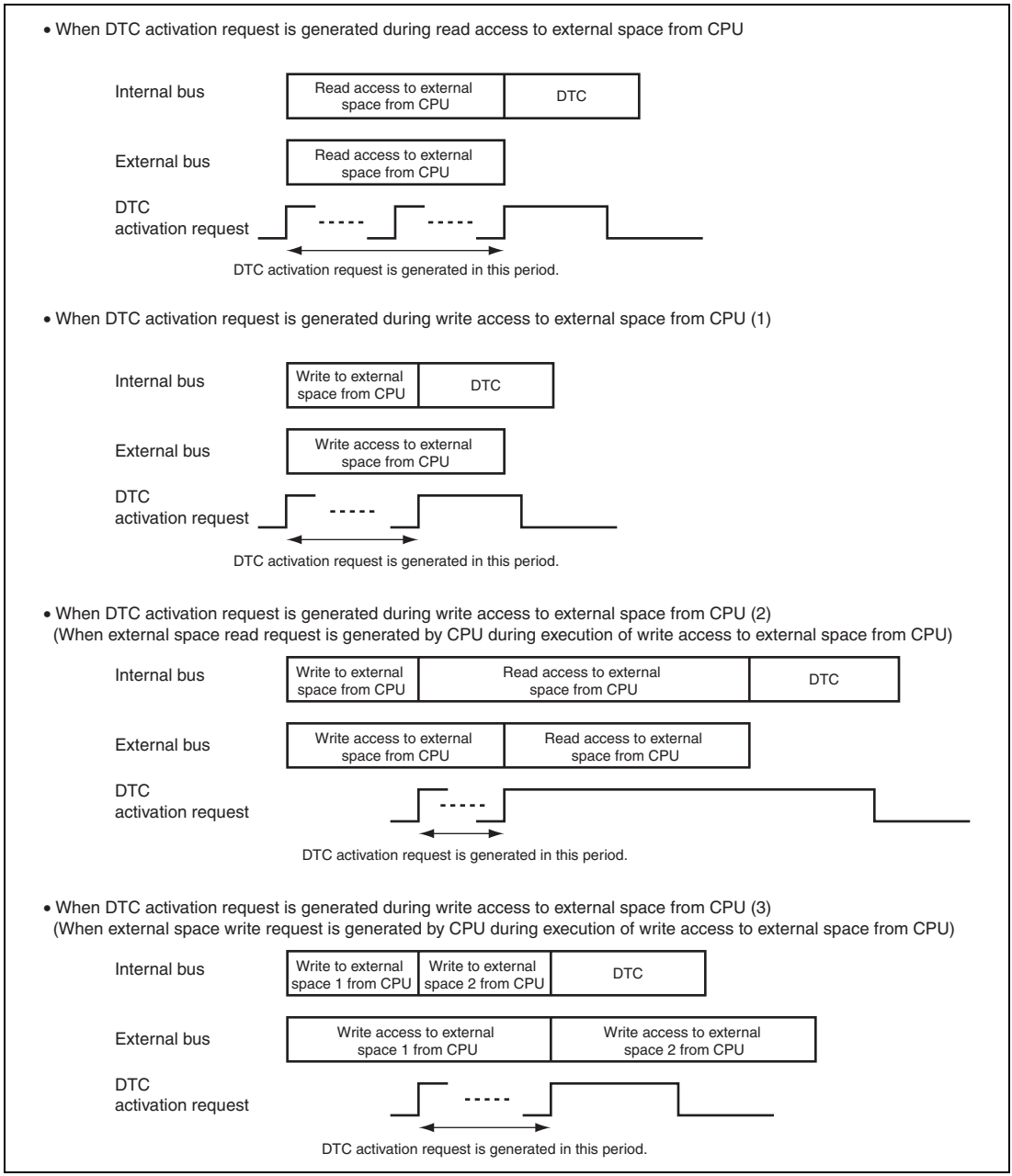



**Figure 9.9 Bus Arbitration When DTC Activation Request Occur during External Space Access from CPU**

The states that do not allow bus arbitration are shown below.

1. Between the read and write cycles of a TAS instruction
2. Multiple bus cycles generated when the data bus width is smaller than the access size (for example, between bus cycles when longword access is made to a memory with a data bus width of 8 bits)

To prevent device malfunction while the bus mastership is transferred to the external device, the LSI negates all of the bus control signals before bus release. When the bus mastership is received, all of the bus control signals are first negated and then driven appropriately. In addition, to prevent noise while the bus control signal is in the high impedance state, pull-up resistors must be connected to these control signals.

Bus mastership is transferred to the external device at the boundary of bus cycles. Namely, bus mastership is released immediately after receiving a bus request when a bus cycle is not being performed. The release of bus mastership is delayed until the bus cycle is complete when a bus cycle is in progress. Even when from outside the LSI it looks like a bus cycle is not being performed, a bus cycle may be performing internally, started by inserting wait cycles between access cycles. Therefore, it cannot be immediately determined whether or not bus mastership has been released by looking at the $\overline{\text{CSn}}$ signal or other bus control signals.

The external bus release by the $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ signal handshaking requires some overhead. If the slave has many tasks, multiple bus cycles should be executed in a bus mastership acquisition. Reducing the cycles required for master to slave bus mastership transitions streamlines the system design.

The LSI has the bus mastership until a bus request is received from the external device. Upon acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LSI releases the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the LSI acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the slave has released the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.

Processing by this LSI continues even while bus mastership is released to an external device, unless an external device is accessed. When an external device is accessed, the LSI enters the state of waiting for bus mastership to be returned.

While the bus is released, sleep mode, software standby mode, and deep software standby mode cannot be entered.

The bus release sequence is as follows. The address bus and data bus are placed in a high-impedance state synchronized with the rising edge of CK. The bus mastership acknowledge signal is asserted 0.5 cycles after the above high impedance state, synchronized with the falling edge of

CK. The bus control signals such as $\overline{\text{CSn}}$ are placed in the high-impedance state at subsequent rising edges of CK. These bus control signals go high one cycle before being placed in the high-impedance state. Bus request signals are sampled at the falling edge of CK.

The sequence for reclaiming the bus mastership from an external device is described below.

At 1.5 cycles after the negation of $\overline{\text{BREQ}}$ is detected at the falling edge of CK, the bus control signals are driven high. The bus acknowledge signal is negated at the next falling edge of the clock. The fastest timing at which actual bus cycles can be resumed after bus control signal assertion is at the rising edge of the CK where address and data signals are driven. Figure 9.10 shows the bus arbitration timing in master mode.

After $\overline{\text{BREQ}}$ assertion (low level; bus request), the $\overline{\text{BREQ}}$ signal should be negated (high level; bus release) only after the $\overline{\text{BACK}}$ is asserted (low level; bus acknowledge). If $\overline{\text{BREQ}}$ is negated before $\overline{\text{BACK}}$ is asserted, $\overline{\text{BACK}}$ may be asserted only for one cycle depending on the $\overline{\text{BREQ}}$ negation timing, and a bus conflict may occur between the external device and this LSI.

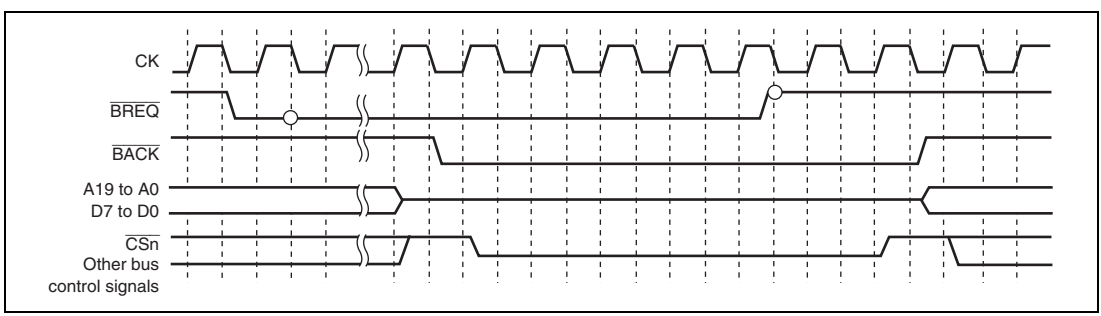



**Figure 9.10 Bus Arbitration Timing**

Acceptance of mastership for the DTC in bus arbitration does not require the insertion of a NOP, so bus access proceeds continuously.

### 9.5.7 Others

#### (1) Reset

The bus state controller (BSC) can be initialized completely only at a power-on reset. At a power-on reset, all signals are negated and output buffers are turned off regardless of the bus cycle state. All control registers are initialized.

In standby, sleep, and manual reset, control registers of the bus state controller are not initialized. At a manual reset, the current bus cycle being executed is completed and then the access wait state is entered. However, a bus arbitration request by the $\overline{\text{BREQ}}$ signal cannot be accepted during manual reset signal assertion.

#### (2) Access in View of LSI Internal Bus Master

There are three types of LSI internal buses: L bus, I bus, and peripheral bus. The CPU is connected to the L bus. The DTC and bus state controller are connected to the I bus. Low-speed peripheral modules are connected to the peripheral bus. On-chip memories are connected bidirectionally to the L bus and I bus.

For an access of an external space or an on-chip peripheral module, the access is initiated via the I bus. Thus, the DTC can be activated without bus arbitration with the CPU while the CPU is accessing an on-chip memory.

Since the bus state controller (BSC) incorporates a one-stage write buffer, the BSC can execute an access via the I bus before the previous external bus cycle is completed in a write cycle. If the on-chip peripheral module is read or written after the external low-speed memory is written, the on-chip peripheral module can be accessed before the completion of the external low-speed memory write cycle.

In read cycles, the CPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the BSC functions in the same way for an access by the DTC.

Since access cannot be performed correctly if any BSC register values are modified while the write buffer is operating, do not modify BSC registers immediately after a write access. If the BSC register need to be modified immediately after a write access, execute dummy read to confirm the completion of the write access, then modify the BSC register.

### 9.5.8 Access to On-Chip FLASH and On-Chip RAM by CPU

Access to the on-chip FLASH for read is synchronized with Iϕ clock and is executed in one clock cycle. For details on programming and erasing, see section 22, Flash Memory.

Access to the on-chip RAM for read/write is synchronized with Iϕ clock and is executed in one clock cycle. For details, see section 23, RAM.

### 9.5.9 Access to On-Chip Peripheral I/O Registers by CPU

Table 9.12 shows the number of cycles required for access to the on-chip peripheral I/O registers by the CPU.

**Table 9.12 Number of Cycles for Access to On-Chip Peripheral I/O Registers**

| | Number of Access Cycles |
|---|---|
| Write | (3 + n) × Iϕ + (1 + m) × Bϕ + 2 × Pϕ |
| Read | (3 + n) × Iϕ + (1 + m) × Bϕ + 2 × Pϕ + 2 × Iϕ |

Notes: 1. When Iϕ:Bϕ = 8:1, n = 0 to 7
When Iϕ:Bϕ = 4:1, n = 0 to 3
When Bϕ:Pϕ = 4:1, m = 0 to 3
When Iϕ:Bϕ = 3:1, n = 0 to 2
When Bϕ:Pϕ = 3:1, m = 0 to 2
When Iϕ:Bϕ = 2:1, n = 0 to 1.
When Bϕ:Pϕ = 2:1, m = 0 to 1.
When Iϕ:Bϕ = 1:1, n = 0.
When Bϕ:Pϕ = 1:1, m = 0
n and m depend on the internal execution state.
2. The clock ratio of MIϕ and MPϕ does not affect the number of access cycles.

Synchronous logic and a layered bus structure have been adopted for this LSI. Data on each bus are input and output in synchronization with rising edges of the corresponding clock signal. The L bus, I bus, and peripheral bus are synchronized with the Iϕ, Bϕ, and Pϕ clock, respectively. Figure 9.11 shows an example of the timing of write access to a register in 2Pϕ cycle access with the connected peripheral bus width of 16 bits when Iϕ:Bϕ:Pϕ = 4:2:2. In access to the on-chip peripheral I/O registers, the CPU requires three cycles of Iϕ for preparation of data transfer to the I bus after the data has been output to the L bus. After these three cycles, data can be transferred to the I bus in synchronization with rising edges of Bϕ. However, as there are two Iϕ clock cycles in a single Bϕ clock cycle when Iϕ: Bϕ = 4:2, transfer of data from the L bus to the I bus takes (3 +

n) × Iϕ (n = 0 to 1) (3 × Iϕ is indicated in figure 9.11). The relation between the timing of data output to the L bus and the rising edge of Bϕ depends on the state of program execution. In the case shown in the figure, where n = 0 and m = 0, the time required for access is 3 × Iϕ + 1 × Bϕ + 2 × Pϕ.

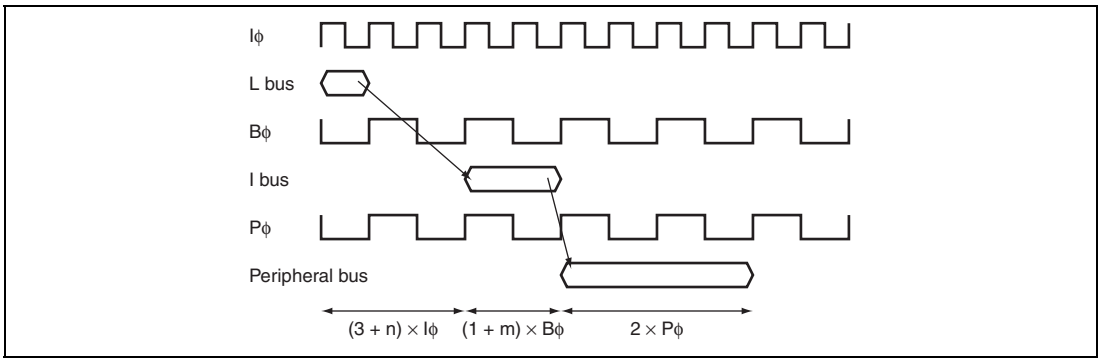



**Figure 9.11 Timing of Write Access to On-Chip Peripheral I/O Registers When Iϕ:Bϕ:Pϕ = 4:2:2**

Figure 9.12 shows an example of timing of read access to the peripheral bus when Iϕ:Bϕ:Pϕ = 4:2:1. Transfer from the L bus to the peripheral bus is performed in the same way as for writing. In the case of reading, however, values output onto the peripheral bus need to be transferred to the CPU. Although transfers from the peripheral bus to the I bus and from the I bus to the L bus are performed in synchronization with the rising edge of the respective bus clocks, a period of 2 × Iϕ is actually required because Iϕ ≥ Bϕ ≥ Pϕ. In the case shown in the figure, where n = 0 and m = 1, the time required for access is 3 × Iϕ + 2 × Bϕ + 2 × Pϕ + 2 × Iϕ.

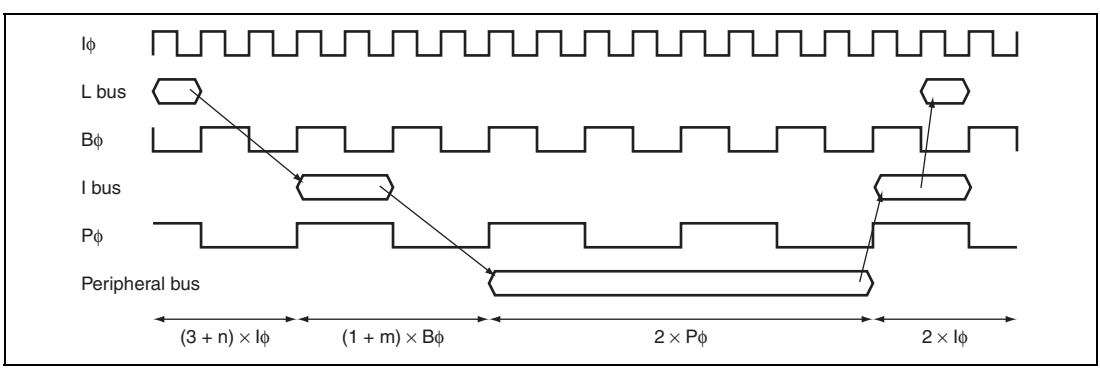



**Figure 9.12 Timing of Read Access to On-Chip Peripheral I/O Registers When Iϕ:Bϕ:Pϕ = 4:2:1**

### 9.5.10 Access to External Memory by CPU

Table 9.13 shows the number of cycles required for access to the external memory by the CPU. As the table shows, the number of cycles varies with the clock ratio, the access size, the external bus width of the LSI, and the setting for wait insertion. For details on the wait-insertion setting, see section 9.4, Register Descriptions.

**Table 9.13 Number of External Access Cycles**

| External Bus Width | Access Size | Write/Read | Number of Access Cycles |
|---|---|---|---|
| 8 bits | Byte | Write | $(1 + n) \times I\phi + (3 + m) \times B\phi$ |
| | | Read | $(1 + n) \times I\phi + (3 + m) \times B\phi + 1 \times I\phi$ |
| | Word | Write | $(1 + n) \times I\phi + (3 + m) \times B\phi + 1 \times (2 + o) \times B\phi$ |
| | | Read | $(1 + n) \times I\phi + (3 + m) \times B\phi + 1 \times (2 + o) \times B\phi + 1 \times I\phi$ |
| | Longword | Write | $(1 + n) \times I\phi + (3 + m) \times B\phi + 3 \times (2 + o) \times B\phi$ |
| | | Read | $(1 + n) \times I\phi + (3 + m) \times B\phi + 3 \times (2 + o) \times B\phi + 1 \times I\phi$ |

Notes: n: When $I\phi:B\phi = 8:1$, n = 0 to 7
When $I\phi:B\phi = 4:1$, n = 0 to 3
When $I\phi:B\phi = 3:1$, n = 0 to 2
When $I\phi:B\phi = 2:1$, n = 0 to 1
When $I\phi:B\phi = 1:1$, n = 0

m, o: m: Wait setting, o: Wait setting + idle setting
For details, see section 9.4, Register Descriptions.

Synchronous logic and a layered bus structure have been adopted for this LSI circuit. Data on each bus are input and output in synchronization with rising edges of the corresponding clock signal. The L bus and I bus are synchronized with the Iϕ and Bϕ clocks, respectively. Figure 9.13 shows an example of the timing of write access to a word of data over the external bus, with a bus-width of 8 bits, when Iϕ:Bϕ = 2:1. Once the CPU has output the data to the L bus, data are transferred to the I bus in synchronization with rising edges of Bϕ. There are two Iϕ clock cycles in a single Bϕ clock cycle when Iϕ: Bϕ = 2:1. Thus, when Iϕ: Bϕ = 2:1, data transfer from the L bus to the I bus takes $(1 + n) \times I\phi$ (n = 0 to 1) ($2 \times I\phi$ is indicated in figure 9.13). The relation between the timing of data output to the L bus and the rising edge of Bϕ depends on the state of program execution. Data output to the I bus are transferred to the external bus after one cycle of Bϕ. External access to each data takes at least two cycles, and this can be prolonged by the BSC register settings (m and o in the formulae for number of access cycles). In the case shown in figure 9.13, since n = 1, m = 0, and o = 0, access takes $2 \times I\phi + 3 \times B\phi + 2 \times B\phi$.

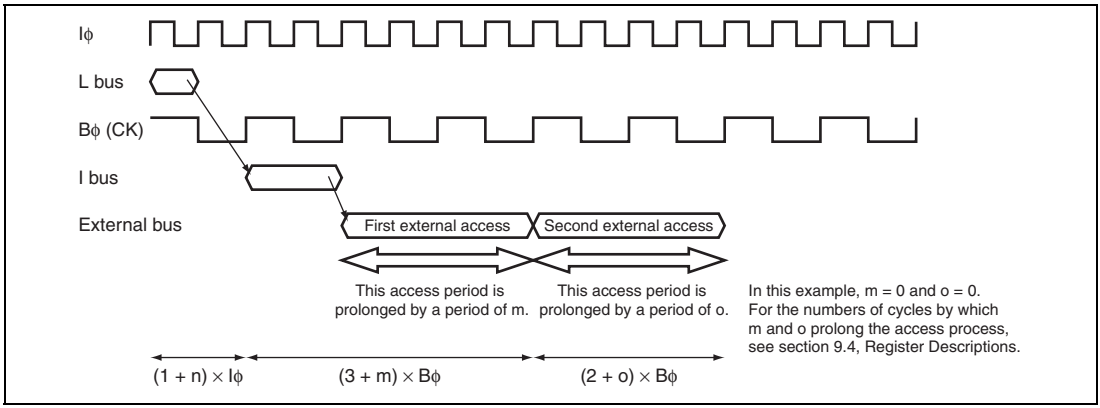



**Figure 9.13 Timing of Write Access to Word Data in External Memory When Iϕ:Bϕ = 2:1 and External Bus Width is 8 Bits**

Figure 9.14 shows an example of the timing of read access when the external bus width is greater than or equal to the data width and Iϕ:Bϕ = 4:1. Transfer from the L bus to the external bus is performed in the same way as for write access. In the case of reading, however, values output onto the external bus must be transferred to the CPU. Transfers from the external bus to the I bus and from the I bus to the L bus are again performed in synchronization with rising edges of the respective bus clocks. In the actual operation, transfer from the external bus to the L bus takes one ϕ period. In the case shown in the figure, where n = 2 and m = 0, access takes 3 × Iϕ + 3 × Bϕ + 1 × Iϕ.

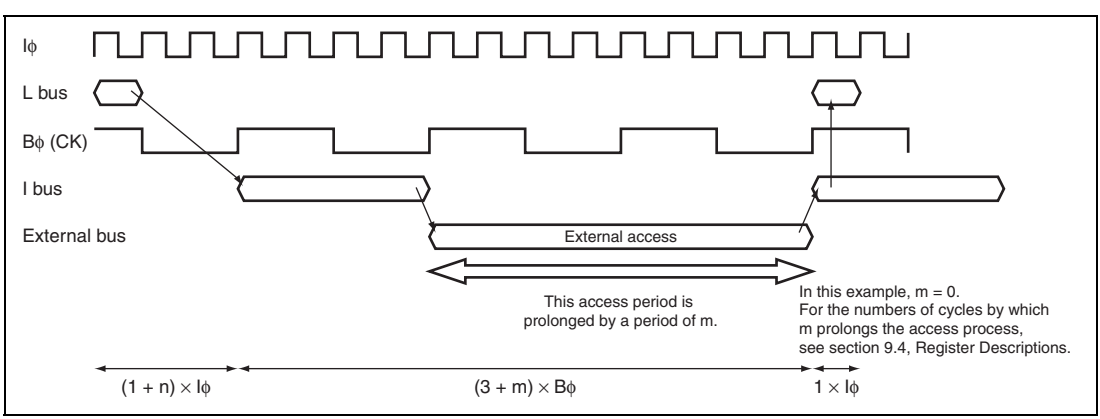



**Figure 9.14 Timing of Read Access with Condition Iϕ:Bϕ = 4:1 and External Bus Width ≥ Data Width**

For access by the DTC, the access cycles are obtained by subtracting the cycles of Iϕ required for L-bus access from the access cycles required for access by the CPU.

# Section 10 Multi-Function Timer Pulse Unit 2 (MTU2)

This LSI has an on-chip multi-function timer pulse unit 2 (MTU2) that comprises six 16-bit timer channels.

## 10.1 Features

- Maximum 16 pulse input/output lines and three pulse input lines
- Selection of eight counter input clocks for each channel (four clocks for channel 5)
- The following operations can be set for channels 0 to 4:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture is possible
  - Register simultaneous input/output is possible by synchronous counter operation
  - A maximum 12-phase PWM output is possible in combination with synchronous operation
- Buffer operation settable for channels 0, 3, and 4
- Phase counting mode settable independently for each of channels 1 and 2
- Cascade connection operation
- Fast access via internal 16-bit bus
- 28 interrupt sources
- Automatic transfer of register data
- A/D converter start trigger can be generated
- Module standby mode can be settable
- A total of six-phase waveform output, which includes complementary PWM output, and positive and negative phases of reset PWM output by interlocking operation of channels 3 and 4, is possible.
- AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset PWM output is settable by interlocking operation of channels 0, 3, and 4, and the selection of two types of waveform outputs (chopping and level) is possible.
- Dead time compensation counter available in channel 5
- In complementary PWM mode, interrupts at the crest and trough of the counter value and A/D converter start triggers can be skipped.

## Table 10.1 MTU2 Functions

| Item | | Channel 0 | Channel 1 | Channel 2 | Channel 3 | Channel 4 | Channel 5 |
|---|---|---|---|---|---|---|---|
| Count clock | | MPϕ/1<br>MPϕ/4<br>MPϕ/16<br>MPϕ/64<br>TCLKA<br>TCLKB<br>TCLKC<br>TCLKD | MPϕ/1<br>MPϕ/4<br>MPϕ/16<br>MPϕ/64<br>MPϕ/256<br>TCLKA<br>TCLKB | MPϕ/1<br>MPϕ/4<br>MPϕ/16<br>MPϕ/64<br>MPϕ/1024<br>TCLKA<br>TCLKB<br>TCLKC | MPϕ/1<br>MPϕ/4<br>MPϕ/16<br>MPϕ/64<br>MPϕ/256<br>MPϕ/1024<br>TCLKA<br>TCLKB | MPϕ/1<br>MPϕ/4<br>MPϕ/16<br>MPϕ/64<br>MPϕ/256<br>MPϕ/1024<br>TCLKA<br>TCLKB | MPϕ/1<br>MPϕ/4<br>MPϕ/16<br>MPϕ/64 |
| General registers | | TGRA_0<br>TGRB_0<br>TGRE_0 | TGRA_1<br>TGRB_1 | TGRA_2<br>TGRB_2 | TGRA_3<br>TGRB_3 | TGRA_4<br>TGRB_4 | TGRU_5<br>TGRV_5<br>TGRW_5 |
| General registers/ buffer registers | | TGRC_0<br>TGRD_0<br>TGRF_0 | — | — | TGRC_3<br>TGRD_3 | TGRC_4<br>TGRD_4 | — |
| I/O pins | | TIOC0A<br>TIOC0B<br>TIOC0C<br>TIOC0D | TIOC1A<br>TIOC1B | TIOC2A<br>TIOC2B | TIOC3A<br>TIOC3B<br>TIOC3C<br>TIOC3D | TIOC4A<br>TIOC4B<br>TIOC4C<br>TIOC4D | Input pins<br>TIC5U<br>TIC5V<br>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 | — | — | — | — | — | √ |
| DTC activation | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture and TCNT overflow or underflow | TGR compare match or input capture |
| A/D converter start trigger | TGRA_0 compare match or input capture<br>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<br>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<br>• Compare match or input capture 0A<br>• Compare match or input capture 0B<br>• Compare match or input capture 0C<br>• Compare match or input capture 0D<br>• Compare match 0E<br>• Compare match 0F<br>• Overflow | 4 sources<br>• Compare match or input capture 1A<br>• Compare match or input capture 1B<br>• Overflow<br>• Underflow | 4 sources<br>• Compare match or input capture 2A<br>• Compare match or input capture 2B<br>• Overflow<br>• Underflow | 5 sources<br>• Compare match or input capture 3A<br>• Compare match or input capture 3B<br>• Compare match or input capture 3C<br>• Compare match or input capture 3D<br>• Overflow | 5 sources<br>• Compare match or input capture 4A<br>• Compare match or input capture 4B<br>• Compare match or input capture 4C<br>• Compare match or input capture 4D<br>• Overflow or underflow | 3 sources<br>• Compare match or input capture 5U<br>• Compare match or input capture 5V<br>• Compare match or input capture 5W |

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

[Legend]

√: Possible

—: Not possible

Figure 10.1 shows a block diagram of the MTU2.

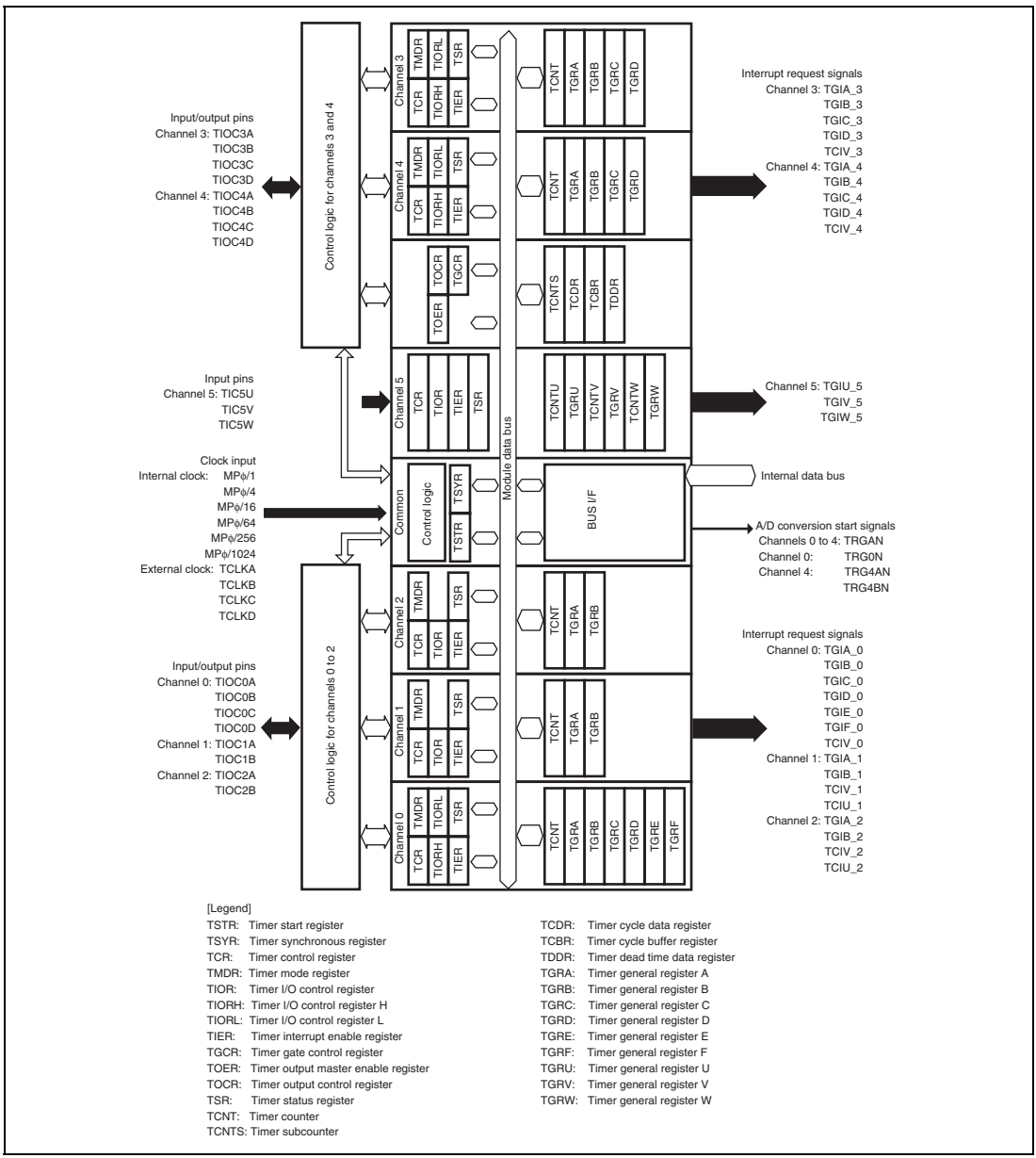



**Figure 10.1 Block Diagram of MTU2**

## 10.2 Input/Output Pins

### Table 10.2 Pin Configuration

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| Common | TCLKA | Input | External clock A input pin<br>(Channel 1 phase counting mode A phase input) |
| | TCLKB | Input | External clock B input pin<br>(Channel 1 phase counting mode B phase input) |
| | TCLKC | Input | External clock C input pin<br>(Channel 2 phase counting mode A phase input) |
| | TCLKD | Input | External clock D input pin<br>(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 |

## 10.3 Register Descriptions

The MTU2 has the following registers. For details on register addresses and register states during each process, refer to section 25, List of Registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 0 is expressed as TCR_0.

**Table 10.3 Register Configuration**

| Register Name | Abbreviation | R/W | Initial value | Address | Access Size |
|---|---|---|---|---|---|
| Timer control register_3 | TCR_3 | R/W | H'00 | H'FFFFC200 | 8, 16, 32 |
| Timer control register_4 | TCR_4 | R/W | H'00 | H'FFFFC201 | 8 |
| Timer mode register_3 | TMDR_3 | R/W | H'00 | H'FFFFC202 | 8, 16 |
| Timer mode register_4 | TMDR_4 | R/W | H'00 | H'FFFFC203 | 8 |
| Timer I/O control register H_3 | TIORH_3 | R/W | H'00 | H'FFFFC204 | 8, 16, 32 |
| Timer I/O control register L_3 | TIORL_3 | R/W | H'00 | H'FFFFC205 | 8 |
| Timer I/O control register H_4 | TIORH_4 | R/W | H'00 | H'FFFFC206 | 8, 16 |
| Timer I/O control register L_4 | TIORL_4 | R/W | H'00 | H'FFFFC207 | 8 |
| Timer interrupt enable register_3 | TIER_3 | R/W | H'00 | H'FFFFC208 | 8, 16 |
| Timer interrupt enable register_4 | TIER_4 | R/W | H'00 | H'FFFFC209 | 8 |
| Timer output master enable register | TOER | R/W | H'C0 | H'FFFFC20A | 8 |
| Timer gate control register | TGCR | R/W | H'80 | H'FFFFC20D | 8 |
| Timer output control register 1 | TOCR1 | R/W | H'00 | H'FFFFC20E | 8, 16 |
| Timer output control register 2 | TOCR2 | R/W | H'00 | H'FFFFC20F | 8 |
| Timer counter_3 | TCNT_3 | R/W | H'0000 | H'FFFFC210 | 16, 32 |
| Timer counter_4 | TCNT_4 | R/W | H'0000 | H'FFFFC212 | 16 |
| Timer cycle data register | TCDR | R/W | H'FFFF | H'FFFFC214 | 16, 32 |
| Timer dead time data register | TDDR | R/W | H'FFFF | H'FFFFC216 | 16 |
| Timer general register A_3 | TGRA_3 | R/W | H'FFFF | H'FFFFC218 | 16, 32 |
| Timer general register B_3 | TGRB_3 | R/W | H'FFFF | H'FFFFC21A | 16 |
| Timer general register A_4 | TGRA_4 | R/W | H'FFFF | H'FFFFC21C | 16, 32 |
| Timer general register B_4 | TGRB_4 | R/W | H'FFFF | H'FFFFC21E | 16 |

| Register Name | Abbreviation | R/W | Initial value | Address | Access Size |
|---|---|---|---|---|---|
| Timer subcounter | TCNTS | R | H'0000 | H'FFFFC220 | 16, 32 |
| Timer cycle buffer register | TCBR | R/W | H'FFFF | H'FFFFC222 | 16 |
| Timer general register C_3 | TGRC_3 | R/W | H'FFFF | H'FFFFC224 | 16, 32 |
| Timer general register D_3 | TGRD_3 | R/W | H'FFFF | H'FFFFC226 | 16 |
| Timer general register C_4 | TGRC_4 | R/W | H'FFFF | H'FFFFC228 | 16, 32 |
| Timer general register D_4 | TGRD_4 | R/W | H'FFFF | H'FFFFC22A | 16 |
| Timer status register_3 | TSR_3 | R/W | H'C0 | H'FFFFC22C | 8, 16 |
| Timer status register_4 | TSR_4 | R/W | H'C0 | H'FFFFC22D | 8 |
| Timer interrupt skipping set register | TITCR | R/W | H'00 | H'FFFFC230 | 8, 16 |
| Timer interrupt skipping counter | TITCNT | R | H'00 | H'FFFFC231 | 8 |
| Timer buffer transfer set register | TBTER | R/W | H'00 | H'FFFFC232 | 8 |
| Timer dead time enable register | TDER | R/W | H'01 | H'FFFFC234 | 8 |
| Timer output level buffer register | TOLBR | R/W | H'00 | H'FFFFC236 | 8 |
| Timer buffer operation transfer mode register_3 | TBTM_3 | R/W | H'00 | H'FFFFC238 | 8, 16 |
| Timer buffer operation transfer mode register_4 | TBTM_4 | R/W | H'00 | H'FFFFC239 | 8 |
| Timer A/D converter start request control register | TADCR | R/W | H'0000 | H'FFFFC240 | 16 |
| Timer A/D converter start request cycle set register A_4 | TADCORA_4 | R/W | H'FFFF | H'FFFFC244 | 16, 32 |
| Timer A/D converter start request cycle set register B_4 | TADCORB_4 | R/W | H'FFFF | H'FFFFC246 | 16 |
| Timer A/D converter start request cycle set buffer register A_4 | TADCOBRA_4 | R/W | H'FFFF | H'FFFFC248 | 16, 32 |
| Timer A/D converter start request cycle set buffer register B_4 | TADCOBRB_4 | R/W | H'FFFF | H'FFFFC24A | 16 |

| Register Name | Abbreviation | R/W | Initial value | Address | Access Size |
|---|---|---|---|---|---|
| Timer waveform control register | TWCR | R/W | H'00 | H'FFFFC260 | 8 |
| Timer start register | TSTR | R/W | H'00 | H'FFFFC280 | 8, 16 |
| Timer synchronous register | TSYR | R/W | H'00 | H'FFFFC281 | 8 |
| Timer counter synchronous start register | TCSYSTR | R/W | H'00 | H'FFFFC282 | 8 |
| Timer read/write enable register | TRWER | R/W | H'01 | H'FFFFC284 | 8 |
| Timer control register_0 | TCR_0 | R/W | H'00 | H'FFFFC300 | 8, 16, 32 |
| Timer mode register_0 | TMDR_0 | R/W | H'00 | H'FFFFC301 | 8 |
| Timer I/O control register H_0 | TIORH_0 | R/W | H'00 | H'FFFFC302 | 8, 16 |
| Timer I/O control register L_0 | TIORL_0 | R/W | H'00 | H'FFFFC303 | 8 |
| Timer interrupt enable register_0 | TIER_0 | R/W | H'00 | H'FFFFC304 | 8, 16, 32 |
| Timer status register_0 | TSR_0 | R/W | H'C0 | H'FFFFC305 | 8 |
| Timer counter_0 | TCNT_0 | R/W | H'0000 | H'FFFFC306 | 16 |
| Timer general register A_0 | TGRA_0 | R/W | H'FFFF | H'FFFFC308 | 16, 32 |
| Timer general register B_0 | TGRB_0 | R/W | H'FFFF | H'FFFFC30A | 16 |
| Timer general register C_0 | TGRC_0 | R/W | H'FFFF | H'FFFFC30C | 16, 32 |
| Timer general register D_0 | TGRD_0 | R/W | H'FFFF | H'FFFFC30E | 16 |
| Timer general register E_0 | TGRE_0 | R/W | H'FFFF | H'FFFFC320 | 16, 32 |
| Timer general register F_0 | TGRF_0 | R/W | H'FFFF | H'FFFFC322 | 16 |
| Timer interrupt enable register 2_0 | TIER2_0 | R/W | H'00 | H'FFFFC324 | 8, 16 |
| Timer status register 2_0 | TSR2_0 | R/W | H'C0 | H'FFFFC325 | 8 |
| Timer buffer operation transfer mode register_0 | TBTM_0 | R/W | H'00 | H'FFFFC326 | 8 |
| Timer control register_1 | TCR_1 | R/W | H'00 | H'FFFFC380 | 8, 16 |
| Timer mode register_1 | TMDR_1 | R/W | H'00 | H'FFFFC381 | 8 |
| Timer I/O control register_1 | TIOR_1 | R/W | H'00 | H'FFFFC382 | 8 |
| Timer interrupt enable register_1 | TIER_1 | R/W | H'00 | H'FFFFC384 | 8, 16, 32 |
| Timer status register_1 | TSR_1 | R/W | H'C0 | H'FFFFC385 | 8 |

| Register Name | Abbreviation | R/W | Initial value | Address | Access Size |
|---|---|---|---|---|---|
| Timer counter_1 | TCNT_1 | R/W | H'0000 | H'FFFFC386 | 16 |
| Timer general register A_1 | TGRA_1 | R/W | H'FFFF | H'FFFFC388 | 16, 32 |
| Timer general register B_1 | TGRB_1 | R/W | H'FFFF | H'FFFFC38A | 16 |
| Timer input capture control register | TICCR | R/W | H'00 | H'FFFFC390 | 8 |
| Timer control register_2 | TCR_2 | R/W | H'00 | H'FFFFC400 | 8, 16 |
| Timer mode register_2 | TMDR_2 | R/W | H'00 | H'FFFFC401 | 8 |
| Timer I/O control register_2 | TIOR_2 | R/W | H'00 | H'FFFFC402 | 8 |
| Timer interrupt enable register_2 | TIER_2 | R/W | H'00 | H'FFFFC404 | 8, 16, 32 |
| Timer status register_2 | TSR_2 | R/W | H'C0 | H'FFFFC405 | 8 |
| Timer counter_2 | TCNT_2 | R/W | H'0000 | H'FFFFC406 | 16 |
| Timer general register A_2 | TGRA_2 | R/W | H'FFFF | H'FFFFC408 | 16, 32 |
| Timer general register B_2 | TGRB_2 | R/W | H'FFFF | H'FFFFC40A | 16 |
| Timer counter U_5 | TCNTU_5 | R/W | H'0000 | H'FFFFC480 | 16, 32 |
| Timer general register U_5 | TGRU_5 | R/W | H'FFFF | H'FFFFC482 | 16 |
| Timer control register U_5 | TCRU_5 | R/W | H'00 | H'FFFFC484 | 8 |
| Timer I/O control register U_5 | TIORU_5 | R/W | H'00 | H'FFFFC486 | 8 |
| Timer counter V_5 | TCNTV_5 | R/W | H'0000 | H'FFFFC490 | 16, 32 |
| Timer general register V_5 | TGRV_5 | R/W | H'FFFF | H'FFFFC492 | 16 |
| Timer control register V_5 | TCRV_5 | R/W | H'00 | H'FFFFC494 | 8 |
| Timer I/O control register V_5 | TIORV_5 | R/W | H'00 | H'FFFFC496 | 8 |
| Timer counter W_5 | TCNTW_5 | R/W | H'0000 | H'FFFFC4A0 | 16, 32 |
| Timer general register W_5 | TGRW_5 | R/W | H'FFFF | H'FFFFC4A2 | 16 |
| Timer control register W_5 | TCRW_5 | R/W | H'00 | H'FFFFC4A4 | 8 |
| Timer I/O control register W_5 | TIORW_5 | R/W | H'00 | H'FFFFC4A6 | 8 |
| Timer status register_5 | TSR_5 | R/W | H'00 | H'FFFFC4B0 | 8 |
| Timer interrupt enable register_5 | TIER_5 | R/W | H'00 | H'FFFFC4B2 | 8 |
| Timer start register_5 | TSTR_5 | R/W | H'00 | H'FFFFC4B4 | 8 |
| Timer compare match clear register | TCNTCMPCLR | R/W | H'00 | H'FFFFC4B6 | 8 |

### 10.3.1 Timer Control Register (TCR)

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

| Bit: | 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<br>These bits select the TCNT counter clearing source. See tables 10.4 and 10.5 for details. |
| 4, 3 | CKEG[1:0] | 00 | R/W | Clock Edge 0 and 1<br>These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. MPϕ/4 both edges = MPϕ/2 rising edge). If phase counting mode is used on channels 1 and 2, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is MPϕ/4 or slower. When MPϕ/1, or the overflow/underflow of another channel is selected for the input clock, although values can be written, counter operation compiles with the initial value.<br>00: Count at rising edge<br>01: Count at falling edge<br>1x: Count at both edges |
| 2 to 0 | TPSC[2:0] | 000 | R/W | Time Prescaler 0 to 2<br>These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 10.6 to 10.10 for details. |

[Legend]

x: Don't care

**Table 10.4 CCLR0 to CCLR2 (Channels 0, 3, and 4)**

| Channel | Bit 7 CCLR2 | Bit 6 CCLR1 | Bit 5 CCLR0 | Description |
|---|---|---|---|---|
| 0, 3, 4 | 0 | 0 | 0 | TCNT clearing disabled |
| | | | 1 | TCNT cleared by TGRA compare match/input capture |
| | | 1 | 0 | TCNT cleared by TGRB compare match/input capture |
| | | | 1 | TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1] |
| | 1 | 0 | 0 | TCNT clearing disabled |
| | | | 1 | TCNT cleared by TGRC compare match/input capture*[2] |
| | | 1 | 0 | TCNT cleared by TGRD compare match/input capture*[2] |
| | | | 1 | TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1] |

Notes: 1. Synchronous operation is set by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

**Table 10.5 CCLR0 to CCLR2 (Channels 1 and 2)**

| Channel | Bit 7 Reserved*[2] | Bit 6 CCLR1 | Bit 5 CCLR0 | Description |
|---|---|---|---|---|
| 1, 2 | 0 | 0 | 0 | TCNT clearing disabled |
| | | | 1 | TCNT cleared by TGRA compare match/input capture |
| | | 1 | 0 | TCNT cleared by TGRB compare match/input capture |
| | | | 1 | TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1] |

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be modified.

**Table 10.6 TPSC0 to TPSC2 (Channel 0)**

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

**Table 10.7 TPSC0 to TPSC2 (Channel 1)**

| Channel | Bit 2 TPSC2 | Bit 1 TPSC1 | Bit 0 TPSC0 | Description |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | Internal clock: counts on MPϕ/1 |
| | | | 1 | Internal clock: counts on MPϕ/4 |
| | | 1 | 0 | Internal clock: counts on MPϕ/16 |
| | | | 1 | Internal clock: counts on MPϕ/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 MPϕ/256 |
| | | | 1 | Counts on TCNT_2 overflow/underflow |

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

**Table 10.8 TPSC0 to TPSC2 (Channel 2)**

| Channel | Bit 2 TPSC2 | Bit 1 TPSC1 | Bit 0 TPSC0 | Description |
|---|---|---|---|---|
| 2 | 0 | 0 | 0 | Internal clock: counts on MPϕ/1 |
| | | | 1 | Internal clock: counts on MPϕ/4 |
| | | 1 | 0 | Internal clock: counts on MPϕ/16 |
| | | | 1 | Internal clock: counts on MPϕ/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 MPϕ/1024 |

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

**Table 10.9 TPSC0 to TPSC2 (Channels 3 and 4)**

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

**Table 10.10 TPSC1 and TPSC0 (Channel 5)**

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

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

### 10.3.2 Timer Mode Register (TMDR)

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode of each channel. The MTU2 has five TMDR registers, one each for channels 0 to 4. TMDR register settings should be changed only when TCNT operation is stopped.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | BFE | BFB | BFA | MD[3:0] | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | - | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | — | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | BFE | 0 | R/W | Buffer Operation E<br>Specifies whether TGRE_0 and TGRF_0 are to operate in the normal way or to be used together for buffer operation. Compare match with TGRF occurs even when TGRF is used as a buffer register.<br>In channels 1 to 4, this bit is reserved. It is always read as 0 and the write value should always be 0.<br>0: TGRE_0 and TGRF_0 operate normally<br>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<br>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 do not take place in modes other than complementary PWM mode, but compare match with TGRD occurs in complementary PWM mode. Since the TGFD flag will be set if a compare match occurs during Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register 3/4 (TIER_3/4) should be cleared to 0.<br>In channels 1 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified.<br>0: TGRB and TGRD operate normally<br>1: TGRB and TGRD used together for buffer operation |
| 4 | BFA | 0 | R/W | Buffer Operation A<br>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 do not take place in modes other than complementary PWM mode, but compare match with TGRC occurs in complementary PWM mode. Since the TGFC flag will be set if a compare match occurs on channel 4 during Tb interval in complementary PWM mode, the TGIEC bit in timer interrupt enable register 4 (TIER_4) should be cleared to 0.<br>In channels 1 and 2, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified.<br>0: TGRA and TGRC operate normally<br>1: TGRA and TGRC used together for buffer operation |
| 3 to 0 | MD[3:0] | 0000 | R/W | Modes 0 to 3<br>These bits are used to set the timer operating mode.<br>See table 10.11 for details. |

**Table 10.11 Setting of Operation Mode by Bits MD0 to MD3**

| Bit 3 MD3 | Bit 2 MD2 | Bit 1 MD1 | Bit 0 MD0 | Description |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Normal operation |
| | | | 1 | Setting prohibited |
| | | 1 | 0 | PWM mode 1 |
| | | | 1 | PWM mode 2*[1] |
| | 1 | 0 | 0 | Phase counting mode 1*[2] |
| | | | 1 | Phase counting mode 2*[2] |
| | | 1 | 0 | Phase counting mode 3*[2] |
| | | | 1 | Phase counting mode 4*[2] |
| 1 | 0 | 0 | 0 | Reset synchronous PWM mode*[3] |
| | | | 1 | Setting prohibited |
| | | 1 | x | Setting prohibited |
| | 1 | 0 | 0 | Setting prohibited |
| | | | 1 | Complementary PWM mode 1 (transmit at crest)*[3] |
| | | 1 | 0 | Complementary PWM mode 2 (transmit at trough)*[3] |
| | | | 1 | Complementary PWM mode 2 (transmit at crest and trough)*[3] |

[Legend]

x: Don't care

Notes: 1. PWM mode 2 cannot be set for channels 3 and 4.

2. Phase counting mode cannot be set for channels 0, 3, and 4.

3. Reset synchronous PWM mode and complementary PWM mode can only be set for channel 3. When channel 3 is set to reset synchronous PWM mode or complementary PWM mode, the channel 4 settings become ineffective and automatically conform to the channel 3 settings. However, do not set channel 4 to reset synchronous PWM mode or complementary PWM mode. Reset synchronous PWM mode and complementary PWM mode cannot be set for channels 0, 1, and 2.

### 10.3.3 Timer I/O Control Register (TIOR)

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

TIOR should be set when TMDR is set to select normal operation, PWM mode, or phase counting mode. The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

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

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIORH_4

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

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

- 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<br>Specify the function of TGRD.<br>See the following tables.<br>TIORL_0: Table 10.13<br>TIORL_3: Table 10.17<br>TIORL_4: Table 10.19 |
| 3 to 0 | IOC[3:0] | 0000 | R/W | I/O Control C0 to C3<br>Specify the function of TGRC.<br>See the following tables.<br>TIORL_0: Table 10.21<br>TIORL_3: Table 10.25<br>TIORL_4: Table 10.27 |

- 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<br>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<br>Specify the function of TGRU_5, TGRV_5, and TGRW_5.<br>For details, see table 10.28. |

**Table 10.12 TIORH_0 (Channel 0)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOB3** | **Bit 6 IOB2** | **Bit 5 IOB1** | **Bit 4 IOB0** | **TGRB_0 Function** | **TIOC0B Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>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<br>Input capture at TCNT_1 count-up/count-down |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.13 TIORL_0 (Channel 0)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOD3** | **Bit 6 IOD2** | **Bit 5 IOD1** | **Bit 4 IOD0** | **TGRD_0 Function** | **TIOC0D Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*[2] | Output retained*[1] |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>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<br>Input capture at TCNT_1 count-up/count-down |

[Legend]

x: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.14 TIOR_1 (Channel 1)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOB3** | **Bit 6 IOB2** | **Bit 5 IOB1** | **Bit 4 IOB0** | **TGRB_1 Function** | **TIOC1B Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |
| | 1 | x | x | | Input capture at generation of TGRC_0 compare match/input capture |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.15 TIOR_2 (Channel 2)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOB3** | **Bit 6 IOB2** | **Bit 5 IOB1** | **Bit 4 IOB0** | **TGRB_2 Function** | **TIOC2B Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.16 TIORH_3 (Channel 3)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOB3** | **Bit 6 IOB2** | **Bit 5 IOB1** | **Bit 4 IOB0** | **TGRB_3 Function** | **TIOC3B Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.17 TIORL_3 (Channel 3)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOD3** | **Bit 6 IOD2** | **Bit 5 IOD1** | **Bit 4 IOD0** | **TGRD_3 Function** | **TIOC3D Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*2 | Output retained*1 |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register*2 | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.18 TIORH_4 (Channel 4)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOB3** | **Bit 6 IOB2** | **Bit 5 IOB1** | **Bit 4 IOB0** | **TGRB_4 Function** | **TIOC4B Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.19 TIORL_4 (Channel 4)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 7 IOD3** | **Bit 6 IOD2** | **Bit 5 IOD1** | **Bit 4 IOD0** | **TGRD_4 Function** | **TIOC4D Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*2 | Output retained*1 |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register*2 | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_4 is set to 1 and TGRD_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.20 TIORH_0 (Channel 0)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOA3** | **Bit 2 IOA2** | **Bit 1 IOA1** | **Bit 0 IOA0** | **TGRA_0 Function** | **TIOC0A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>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<br>Input capture at TCNT_1 count-up/count-down |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.21 TIORL_0 (Channel 0)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOC3** | **Bit 2 IOC2** | **Bit 1 IOC1** | **Bit 0 IOC0** | **TGRC_0 Function** | **TIOC0C Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*2 | Output retained*1 |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>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<br>Input capture at TCNT_1 count-up/count-down |

[Legend]

x: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.22 TIOR_1 (Channel 1)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOA3** | **Bit 2 IOA2** | **Bit 1 IOA1** | **Bit 0 IOA0** | **TGRA_1 Function** | **TIOC1A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |
| | 1 | x | x | | Input capture at generation of channel 0/TGRA_0 compare match/input capture |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.23 TIOR_2 (Channel 2)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOA3** | **Bit 2 IOA2** | **Bit 1 IOA1** | **Bit 0 IOA0** | **TGRA_2 Function** | **TIOC2A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.24 TIORH_3 (Channel 3)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOA3** | **Bit 2 IOA2** | **Bit 1 IOA1** | **Bit 0 IOA0** | **TGRA_3 Function** | **TIOC3A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.25 TIORL_3 (Channel 3)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOC3** | **Bit 2 IOC2** | **Bit 1 IOC1** | **Bit 0 IOC0** | **TGRC_3 Function** | **TIOC3C Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*2 | Output retained*1 |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register*2 | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.26 TIORH_4 (Channel 4)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOA3** | **Bit 2 IOA2** | **Bit 1 IOA1** | **Bit 0 IOA0** | **TGRA_4 Function** | **TIOC4A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output retained* |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

**Table 10.27 TIORL_4 (Channel 4)**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOC3** | **Bit 2 IOC2** | **Bit 1 IOC1** | **Bit 0 IOC0** | **TGRC_4 Function** | **TIOC4C Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*2 | Output retained*1 |
| | | | 1 | | Initial output is 0<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 0<br>1 output at compare match |
| | | | 1 | | Initial output is 0<br>Toggle output at compare match |
| | 1 | 0 | 0 | | Output retained |
| | | | 1 | | Initial output is 1<br>0 output at compare match |
| | | 1 | 0 | | Initial output is 1<br>1 output at compare match |
| | | | 1 | | Initial output is 1<br>Toggle output at compare match |
| 1 | x | 0 | 0 | Input capture register*2 | Input capture at rising edge |
| | | | 1 | | Input capture at falling edge |
| | | 1 | x | | Input capture at both edges |

[Legend]

x: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_4 is set to 1 and TGRC_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.28 TIORU_5, TIORV_5, and TIORW_5 (Channel 5)**

| | | | | | Description | |
|---|---|---|---|---|---|---|
| **Bit 4 IOC4** | **Bit 3 IOC3** | **Bit 2 IOC2** | **Bit 1 IOC1** | **Bit 0 IOC0** | **TGRU_5, TGRV_5, and TGRW_5 Function** | **TIC5U, TIC5V, and TIC5W Pin Function** |
| 0 | 0 | 0 | 0 | 0 | Compare match register | Compare match |
| | | | | 1 | | Setting prohibited |
| | | | 1 | x | | Setting prohibited |
| | | 1 | x | x | | Setting prohibited |
| | 1 | x | x | x | | Setting prohibited |
| 1 | 0 | 0 | 0 | 0 | Input capture register | Setting prohibited |
| | | | | 1 | | Input capture at rising edge |
| | | | 1 | 0 | | Input capture at falling edge |
| | | | | 1 | | Input capture at both edges |
| | | 1 | x | x | | Setting prohibited |
| | 1 | 0 | 0 | 0 | | Setting prohibited |
| | | | | 1 | | Measurement of low pulse width of external input signal<br>Capture at trough of complementary PWM mode |
| | | | 1 | 0 | | Measurement of low pulse width of external input signal<br>Capture at crest of complementary PWM mode |
| | | | | 1 | | Measurement of low pulse width of external input signal<br>Capture at crest and trough of complementary PWM mode |
| | | 1 | 0 | 0 | | Setting prohibited |
| | | | | 1 | | Measurement of high pulse width of external input signal<br>Capture at trough of complementary PWM mode |
| | | | 1 | 0 | | Measurement of high pulse width of external input signal<br>Capture at crest of complementary PWM mode |
| | | | | 1 | | Measurement of high pulse width of external input signal<br>Capture at crest and trough of complementary PWM mode |

[Legend]

x: Don't care

### 10.3.4 Timer Compare Match Clear Register (TCNTCMPCLR)

TCNTCMPCLR is an 8-bit readable/writable register that specifies requests to clear TCNTU_5, TCNTV_5, and TCNTW_5. The MTU2 has one TCNTCMPCLR in channel 5.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | CMPCLR5U | CMPCLR5V | CMPCLR5W |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 2 | CMPCLR5U | 0 | R/W | TCNT Compare Clear 5U<br>Enables or disables requests to clear TCNTU_5 at TCNTU_5 and TGRU_5 compare match or input capture.<br>0: Disables TCNTU_5 to be cleared to H'0000 at TCNTU_5 and TGRU_5 compare match or input capture<br>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<br>Enables or disables requests to clear TCNTV_5 at TCNTV_5 and TGRV_5 compare match or input capture.<br>0: Disables TCNTV_5 to be cleared to H'0000 at TCNTV_5 and TGRV_5 compare match or input capture<br>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<br>Enables or disables requests to clear TCNTW_5 at TGRW_5 compare match or input capture.<br>0: Disables TCNTW_5 to be cleared to H'0000 at TCNTW_5 and TGRW_5 compare match or input capture<br>1: Enables TCNTW_5 to be cleared to H'0000 at TCNTW_5 and TGRW_5 compare match or input capture |

### 10.3.5 Timer Interrupt Enable Register (TIER)

The TIER registers are 8-bit readable/writable registers that control enabling or disabling of interrupt requests for each channel. The MTU2 has seven TIER registers, two for channel 0 and one each for channels 1 to 5.

- TIER_0, TIER_1, TIER_2, TIER_3, TIER_4

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TTGE | TTGE2 | TCIEU | TCIEV | TGIED | TGIEC | TGIEB | TGIEA |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TTGE | 0 | R/W | A/D Converter Start Request Enable<br>Enables or disables generation of A/D converter start requests by TGRA input capture/compare match.<br>0: A/D converter start request generation disabled<br>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<br>Enables or disables generation of A/D converter start requests by TCNT_4 underflow (trough) in complementary PWM mode.<br>In channels 0 to 3, bit 6 is reserved. It is always read as 0 and the write value should always be 0.<br>0: A/D converter start request generation by TCNT_4 underflow (trough) disabled<br>1: A/D converter start request generation by TCNT_4 underflow (trough) enabled |
| 5 | TCIEU | 0 | R/W | Underflow Interrupt Enable<br>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.<br>In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0.<br>0: Interrupt requests (TCIU) by TCFU disabled<br>1: Interrupt requests (TCIU) by TCFU enabled |
| 4 | TCIEV | 0 | R/W | Overflow Interrupt Enable<br>Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1.<br>0: Interrupt requests (TCIV) by TCFV disabled<br>1: Interrupt requests (TCIV) by TCFV enabled |
| 3 | TGIED | 0 | R/W | TGR Interrupt Enable D<br>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.<br>In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 0.<br>0: Interrupt requests (TGID) by TGFD bit disabled<br>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<br>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.<br>In channels 1 and 2, bit 2 is reserved. It is always read as 0 and the write value should always be 0.<br>0: Interrupt requests (TGIC) by TGFC bit disabled<br>1: Interrupt requests (TGIC) by TGFC bit enabled |
| 1 | TGIEB | 0 | R/W | TGR Interrupt Enable B<br>Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1.<br>0: Interrupt requests (TGIB) by TGFB bit disabled<br>1: Interrupt requests (TGIB) by TGFB bit enabled |
| 0 | TGIEA | 0 | R/W | TGR Interrupt Enable A<br>Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1.<br>0: Interrupt requests (TGIA) by TGFA bit disabled<br>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<br>Enables or disables generation of A/D converter start requests by compare match between TCNT_0 and TGRE_0.<br>0: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 disabled<br>1: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 enabled |
| 6 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | TGIEF | 0 | R/W | TGR Interrupt Enable F<br>Enables or disables interrupt requests by compare match between TCNT_0 and TGRF_0.<br>0: Interrupt requests (TGIF) by TGFE bit disabled<br>1: Interrupt requests (TGIF) by TGFE bit enabled |
| 0 | TGIEE | 0 | R/W | TGR Interrupt Enable E<br>Enables or disables interrupt requests by compare match between TCNT_0 and TGRE_0.<br>0: Interrupt requests (TGIE) by TGEE bit disabled<br>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<br>These bits are always read as 0. The write value should always be 0. |
| 2 | TGIE5U | 0 | R/W | TGR Interrupt Enable 5U<br>Enables or disables interrupt requests (TGIU_5) by the CMFU5 bit when the CMFU5 bit in TSR_5 is set to 1.<br>0: Interrupt requests (TGIU_5) disabled<br>1: Interrupt requests (TGIU_5) enabled |
| 1 | TGIE5V | 0 | R/W | TGR Interrupt Enable 5V<br>Enables or disables interrupt requests (TGIV_5) by the CMFV5 bit when the CMFV5 bit in TSR_5 is set to 1.<br>0: Interrupt requests (TGIV_5) disabled<br>1: Interrupt requests (TGIV_5) enabled |
| 0 | TGIE5W | 0 | R/W | TGR Interrupt Enable 5W<br>Enables or disables interrupt requests (TGIW_5) by the CMFW5 bit when the CMFW5 bit in TSR_5 is set to 1.<br>0: Interrupt requests (TGIW_5) disabled<br>1: Interrupt requests (TGIW_5) enabled |

### 10.3.6 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. The MTU2 has seven TSR registers, two for channel 0 and one each for channels 1 to 5.

- TSR_0, TSR_1, TSR_2, TSR_3, TSR_4

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCFD | - | TCFU | TCFV | TGFD | TGFC | TGFB | TGFA |
| Initial value: | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R/(W)*1 | R/(W)*1 | R/(W)*1 | R/(W)*1 | R/(W)*1 | R/(W)*1 |

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TCFD | 1 | R | Count Direction Flag<br>Status flag that shows the direction in which TCNT counts in channels 1 to 4.<br>In channel 0, bit 7 is reserved. It is always read as 1 and the write value should always be 1.<br>0: TCNT counts down<br>1: TCNT counts up |
| 6 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 5 | TCFU | 0 | R/(W)*1 | Underflow Flag<br>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.<br>In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0.<br>[Setting condition]<br>• When the TCNT value underflows (changes from H'0000 to H'FFFF)<br>[Clearing condition]<br>• 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<br>Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing.<br>[Setting condition]<br>• When the TCNT value overflows (changes from H'FFFF to H'0000)<br>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.<br>[Clearing condition]<br>• When 0 is written to TCFV after reading TCFV = 1*2<br>In channel 4, when DTC is activated by TCIV interrupt and the DISEL bit of MRB in DTC is 0, this flag is also cleared. |
| 3 | TGFD | 0 | R/(W)*1 | Input Capture/Output Compare Flag D<br>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.<br>[Setting conditions]<br>• When TCNT = TGRD and TGRD is functioning as output compare register<br>• When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register<br>[Clearing conditions]<br>• When DTC is activated by TGID interrupt and the DISEL bit of MRB in DTC is 0<br>• 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<br>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.<br>[Setting conditions]<br>• When TCNT = TGRC and TGRC is functioning as output compare register<br>• When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register<br>[Clearing conditions]<br>• When DTC is activated by TGIC interrupt and the DISEL bit of MRB in DTC is 0<br>• When 0 is written to TGFC after reading TGFC = 1*[2] |
| 1 | TGFB | 0 | R/(W)*[1] | Input Capture/Output Compare Flag B<br>Status flag that indicates the occurrence of TGRB input capture or compare match. Only 0 can be written, for flag clearing.<br>[Setting conditions]<br>• When TCNT = TGRB and TGRB is functioning as output compare register<br>• When TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register<br>[Clearing conditions]<br>• When DTC is activated by TGIB interrupt and the DISEL bit of MRB in DTC is 0<br>• 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<br>Status flag that indicates the occurrence of TGRA input capture or compare match. Only 0 can be written, for flag clearing.<br>[Setting conditions]<br>• When TCNT = TGRA and TGRA is functioning as output compare register<br>• When TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register<br>[Clearing conditions]<br>• When DTC is activated by TGIA interrupt and the DISEL bit of MRB in DTC is 0<br>• 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. If another flag setting condition occurs before writing 0 to the bit after reading it as 1, the flag will not be cleared by writing 0 to it once. In this case, read the bit as 1 again and write 0 to it.

- 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<br>These bits are always read as 1. The write value should always be 1. |
| 5 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | TGFF | 0 | R/(W)*1 | Compare Match Flag F<br>Status flag that indicates the occurrence of compare match between TCNT_0 and TGRF_0.<br>[Setting condition]<br>• When TCNT_0 = TGRF_0 and TGRF_0 is functioning as compare register<br>[Clearing condition]<br>• When 0 is written to TGFF after reading TGFF = 1*2 |
| 0 | TGFE | 0 | R/(W)*1 | Compare Match Flag E<br>Status flag that indicates the occurrence of compare match between TCNT_0 and TGRE_0.<br>[Setting condition]<br>• When TCNT_0 = TGRE_0 and TGRE_0 is functioning as compare register<br>[Clearing condition]<br>• 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. If another flag setting condition occurs before writing 0 to the bit after reading it as 1, the flag will not be cleared by writing 0 to it once. In this case, read the bit as 1 again and write 0 to it.

- 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<br>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<br>Status flag that indicates the occurrence of TGRU_5 input capture or compare match.<br>[Setting conditions]<br>• When TCNTU_5 = TGRU_5 and TGRU_5 is functioning as output compare register<br>• When TCNTU_5 value is transferred to TGRU_5 by input capture signal and TGRU_5 is functioning as input capture register<br>• When TCNTU_5 value is transferred to TGRU_5 and TGRU_5 is functioning as a register for measuring the pulse width of the external input signal. The transfer timing is specified by the IOC bits in timer I/O control register U_5 (TIORU_5)*2<br>[Clearing conditions]<br>• When DTC is activated by a TGIU_5 interrupt and the DISEL bit of MRB in DTC is 0<br>• 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<br>Status flag that indicates the occurrence of TGRV_5 input capture or compare match.<br>[Setting conditions]<br>• When TCNTV_5 = TGRV_5 and TGRV_5 is functioning as output compare register<br>• When TCNTV_5 value is transferred to TGRV_5 by input capture signal and TGRV_5 is functioning as input capture register<br>• When TCNTV_5 value is transferred to TGRV_5 and TGRV_5 is functioning as a register for measuring the pulse width of the external input signal. The transfer timing is specified by the IOC bits in timer I/O control register V_5 (TIORV_5)*[2]<br>[Clearing conditions]<br>• When DTC is activated by a TGIV_5 interrupt and the DISEL bit of MRB in DTC is 0<br>• When 0 is written to CMFV5 after reading CMFV5 = 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | CMFW5 | 0 | R/(W)*[1] | Compare Match/Input Capture Flag W5<br>Status flag that indicates the occurrence of TGRW_5 input capture or compare match.<br>[Setting conditions]<br>• When TCNTW_5 = TGRW_5 and TGRW_5 is functioning as output compare register<br>• When TCNTW_5 value is transferred to TGRW_5 by input capture signal and TGRW_5 is functioning as input capture register<br>• When TCNTW_5 value is transferred to TGRW_5 and TGRW_5 is functioning as a register for measuring the pulse width of the external input signal. The transfer timing is specified by the IOC bits in timer I/O control register W_5 (TIORW_5)*[2]<br>[Clearing conditions]<br>• When DTC is activated by a TGIW_5 interrupt and the DISEL bit of MRB in DTC is 0<br>• 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. The transfer timing is specified by the IOC bit in timer I/O control registers U_5/V_5/W_5 (TIORU_5, TIORV_5, TIORW_5).

### 10.3.7 Timer Buffer Operation Transfer Mode Register (TBTM)

The TBTM registers are 8-bit readable/writable registers that specify the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU2 has three TBTM registers, one each for channels 0, 3, and 4.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | TTSE | TTSB | TTSA |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 2 | TTSE | 0 | R/W | Timing Select E<br>Specifies the timing for transferring data from TGRF_0 to TGRE_0 when they are used together for buffer operation.<br>In channels 3 and 4, bit 2 is reserved. It is always read as 0 and the write value should always be 0. When using channel 0 in other than PWM mode, do not set this bit to 1.<br>0: When compare match E occurs in channel 0<br>1: When TCNT_0 is cleared |
| 1 | TTSB | 0 | R/W | Timing Select B<br>Specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When using a channel in other than PWM mode, do not set this bit to1.<br>0: When compare match B occurs in each channel<br>1: When TCNT is cleared in each channel |
| 0 | TTSA | 0 | R/W | Timing Select A<br>Specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When using a channel in other than PWM mode, do not set this bit to 1.<br>0: When compare match A occurs in each channel<br>1: When TCNT is cleared in each channel |

### 10.3.8 Timer Input Capture Control Register (TICCR)

TICCR is an 8-bit readable/writable register that specifies input capture conditions when TCNT_1 and TCNT_2 are cascaded. The MTU2 has one TICCR in channel 1.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | I2BE | I2AE | I1BE | I1AE |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 3 | I2BE | 0 | R/W | Input Capture Enable<br>Specifies whether to include the TIOC2B pin in the TGRB_1 input capture conditions.<br>0: Does not include the TIOC2B pin in the TGRB_1 input capture conditions<br>1: Includes the TIOC2B pin in the TGRB_1 input capture conditions |
| 2 | I2AE | 0 | R/W | Input Capture Enable<br>Specifies whether to include the TIOC2A pin in the TGRA_1 input capture conditions.<br>0: Does not include the TIOC2A pin in the TGRA_1 input capture conditions<br>1: Includes the TIOC2A pin in the TGRA_1 input capture conditions |
| 1 | I1BE | 0 | R/W | Input Capture Enable<br>Specifies whether to include the TIOC1B pin in the TGRB_2 input capture conditions.<br>0: Does not include the TIOC1B pin in the TGRB_2 input capture conditions<br>1: Includes the TIOC1B pin in the TGRB_2 input capture conditions |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | I1AE | 0 | R/W | Input Capture Enable<br>Specifies whether to include the TIOC1A pin in the TGRA_2 input capture conditions.<br>0: Does not include the TIOC1A pin in the TGRA_2 input capture conditions<br>1: Includes the TIOC1A pin in the TGRA_2 input capture conditions |

### 10.3.9 Timer Synchronous Clear Register (TSYCR)

TSYCR is an 8-bit readable/writable register that specifies conditions for clearing TCNT_3 and TCNT_4 in the MTU2S in synchronization with the MTU2. The MTU2S has one TSYCR in channel 3 but the MTU2 has no TSYCR.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CE0A | CE0B | CE0C | CE0D | CE1A | CE1B | CE2A | CE2B |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CE0A | 0 | R/W | Clear Enable 0A<br>Enables or disables counter clearing when the TGFA flag of TSR_0 in the MTU2 is set.<br>0: Disables counter clearing by the TGFA flag in TSR_0<br>1: Enables counter clearing by the TGFA flag in TSR_0 |
| 6 | CE0B | 0 | R/W | Clear Enable 0B<br>Enables or disables counter clearing when the TGFB flag of TSR_0 in the MTU2 is set.<br>0: Disables counter clearing by the TGFB flag in TSR_0<br>1: Enables counter clearing by the TGFB flag in TSR_0 |
| 5 | CE0C | 0 | R/W | Clear Enable 0C<br>Enables or disables counter clearing when the TGFC flag of TSR_0 in the MTU2 is set.<br>0: Disables counter clearing by the TGFC flag in TSR_0<br>1: Enables counter clearing by the TGFC flag in TSR_0 |
| 4 | CE0D | 0 | R/W | Clear Enable 0D<br>Enables or disables counter clearing when the TGFD flag of TSR_0 in the MTU2 is set.<br>0: Disables counter clearing by the TGFD flag in TSR_0<br>1: Enables counter clearing by the TGFD flag in TSR_0 |
| 3 | CE1A | 0 | R/W | Clear Enable 1A<br>Enables or disables counter clearing when the TGFA flag of TSR_1 in the MTU2 is set.<br>0: Disables counter clearing by the TGFA flag in TSR_1<br>1: Enables counter clearing by the TGFA flag in TSR_1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | CE1B | 0 | R/W | Clear Enable 1B<br>Enables or disables counter clearing when the TGFB flag of TSR_1 in the MTU2 is set.<br>0: Disables counter clearing by the TGFB flag in TSR_1<br>1: Enables counter clearing by the TGFB flag in TSR_1 |
| 1 | CE2A | 0 | R/W | Clear Enable 2A<br>Enables or disables counter clearing when the TGFA flag of TSR_2 in the MTU2 is set.<br>0: Disables counter clearing by the TGFA flag in TSR_2<br>1: Enables counter clearing by the TGFA flag in TSR_2 |
| 0 | CE2B | 0 | R/W | Clear Enable 2B<br>Enables or disables counter clearing when the TGFB flag of TSR_2 in the MTU2 is set.<br>0: Disables counter clearing by the TGFB flag in TSR_2<br>1: Enables counter clearing by the TGFB flag in TSR_2 |

### 10.3.10 Timer A/D Converter Start Request Control Register (TADCR)

TADCR is a 16-bit readable/writable register that enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation. The MTU2 has one TADCR in channel 4.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BF[1:0] | | - | - | - | - | - | - | UT4AE | DT4AE | UT4BE | DT4BE | ITA3AE | ITA4VE | ITB3AE | ITB4VE |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0* | 0 | 0* | 0* | 0* | 0* | 0* |
| R/W: | R/W | R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note: * Do not set to 1 when complementary PWM mode is not selected.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15, 14 | BF[1:0] | 00 | R/W | TADCOBRA_4/TADCOBRB_4 Transfer Timing Select<br>Select the timing for transferring data from TADCOBRA_4 and TADCOBRB_4 to TADCORA_4 and TADCORB_4.<br>For details, see table 10.29. |
| 13 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 7 | UT4AE | 0 | R/W | Up-Count TRG4AN Enable<br>Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 up-count operation.<br>0: A/D converter start requests (TRG4AN) disabled during TCNT_4 up-count operation<br>1: A/D converter start requests (TRG4AN) enabled during TCNT_4 up-count operation |
| 6 | DT4AE | 0* | R/W | Down-Count TRG4AN Enable<br>Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 down-count operation.<br>0: A/D converter start requests (TRG4AN) disabled during TCNT_4 down-count operation<br>1: A/D converter start requests (TRG4AN) enabled during TCNT_4 down-count operation |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | UT4BE | 0 | R/W | Up-Count TRG4BN Enable<br>Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 up-count operation.<br>0: A/D converter start requests (TRG4BN) disabled during TCNT_4 up-count operation<br>1: A/D converter start requests (TRG4BN) enabled during TCNT_4 up-count operation |
| 4 | DT4BE | 0* | R/W | Down-Count TRG4BN Enable<br>Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 down-count operation.<br>0: A/D converter start requests (TRG4BN) disabled during TCNT_4 down-count operation<br>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<br>Select whether to link A/D converter start requests (TRG4AN) with TGIA_3 interrupt skipping operation.<br>0: Does not link with TGIA_3 interrupt skipping<br>1: Links with TGIA_3 interrupt skipping |
| 2 | ITA4VE | 0* | R/W | TCIV_4 Interrupt Skipping Link Enable<br>Select whether to link A/D converter start requests (TRG4AN) with TCIV_4 interrupt skipping operation.<br>0: Does not link with TCIV_4 interrupt skipping<br>1: Links with TCIV_4 interrupt skipping |
| 1 | ITB3AE | 0* | R/W | TGIA_3 Interrupt Skipping Link Enable<br>Select whether to link A/D converter start requests (TRG4BN) with TGIA_3 interrupt skipping operation.<br>0: Does not link with TGIA_3 interrupt skipping<br>1: Links with TGIA_3 interrupt skipping |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | ITB4VE | 0* | R/W | TCIV_4 Interrupt Skipping Link Enable<br>Select whether to link A/D converter start requests (TRG4BN) with TCIV_4 interrupt skipping operation.<br>0: Does not link with TCIV_4 interrupt skipping<br>1: Links with TCIV_4 interrupt skipping |

Notes: 1. TADCR must not be accessed in eight bits; it should always be accessed in 16 bits.
2. When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).
3. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
* Do not set to 1 when complementary PWM mode is not selected.

**Table 10.29 Setting of Transfer Timing by 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.

### 10.3.11 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4)

TADCORA_4 and TADCORB_4 are 16-bit readable/writable registers. When the TCNT_4 count reaches the value in TADCORA_4 or TADCORB_4, a corresponding A/D converter start request will be issued.

TADCORA_4 and TADCORB_4 are initialized to H'FFFF.

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

Note: TADCORA_4 and TADCORB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

### 10.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4)

TADCOBRA_4 and TADCOBRB_4 are 16-bit readable/writable registers. When the crest or trough of the TCNT_4 count is reached, these register values are transferred to TADCORA_4 and TADCORB_4, respectively.

TADCOBRA_4 and TADCOBRB_4 are initialized to H'FFFF.

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

Note: TADCOBRA_4 and TADCOBRB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

### 10.3.13 Timer Counter (TCNT)

The TCNT counters are 16-bit readable/writable counters. The MTU2 has eight TCNT counters, one each for channels 0 to 4 and three (TCNTU_5, TCNTV_5, and TCNTW_5) for channel 5.

The TCNT counters are initialized to H'0000 by a reset.

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

Note: The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

### 10.3.14 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers. The MTU2 has 21 TGR registers, six for channel 0, two each for channels 1 and 2, four each for channels 3 and 4, and three for channel 5.

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD for channels 0, 3, and 4 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

TGRE_0 and TGRF_0 function as compare registers. When the TCNT_0 count matches the TGRE_0 value, an A/D converter start request can be issued. TGRF can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.

TGRU_5, TGRV_5, and TGRW_5 function as compare match, input capture, or external pulse width measurement registers.

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

Note: The TGR registers must not be accessed in eight bits; they should always be accessed in 16 bits.
TGR registers are initialized to H'FFFF.

### 10.3.15 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage of TCNT for channels 0 to 4.

TSTR_5 is an 8-bit readable/writable register that selects operation/stoppage of TCNTU_5, TCNTV_5, and TCNTW_5 for channel 5.

When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

- TSTR

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CST4 | CST3 | - | - | - | CST2 | CST1 | CST0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R | R | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CST4 | 0 | R/W | Counter Start 4 and 3<br>These bits select operation or stoppage for TCNT.<br>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.<br>0: TCNT_4 and TCNT_3 count operation is stopped<br>1: TCNT_4 and TCNT_3 performs count operation |
| 6 | CST3 | 0 | R/W | |
| 5 to 3 | — | All 0 | R | Reserved<br>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<br>These bits select operation or stoppage for TCNT.<br>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.<br>0: TCNT_2 to TCNT_0 count operation is stopped<br>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<br>These bits are always read as 0. The write value should always be 0. |
| 2 | CSTU5 | 0 | R/W | Counter Start U5<br>Selects operation or stoppage for TCNTU_5.<br>0: TCNTU_5 count operation is stopped<br>1: TCNTU_5 performs count operation |
| 1 | CSTV5 | 0 | R/W | Counter Start V5<br>Selects operation or stoppage for TCNTV_5.<br>0: TCNTV_5 count operation is stopped<br>1: TCNTV_5 performs count operation |
| 0 | CSTW5 | 0 | R/W | Counter Start W5<br>Selects operation or stoppage for TCNTW_5.<br>0: TCNTW_5 count operation is stopped<br>1: TCNTW_5 performs count operation |

### 10.3.16 Timer Synchronous Register (TSYR)

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 4 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SYNC4 | SYNC3 | - | - | - | SYNC2 | SYNC1 | SYNC0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R | R | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | SYNC4 | 0 | R/W | Timer Synchronous operation 4 and 3 |
| 6 | SYNC3 | 0 | R/W | These bits are used to select whether operation is independent of or synchronized with other channels.<br>When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible.<br>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.<br>0: TCNT_4 and TCNT_3 operate independently (TCNT presetting/clearing is unrelated to other channels)<br>1: TCNT_4 and TCNT_3 performs synchronous operation<br>TCNT synchronous presetting/synchronous clearing is possible |
| 5 to 3 | — | All 0 | R | Reserved<br>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<br>These bits are used to select whether operation is independent of or synchronized with other channels.<br>When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible.<br>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.<br>0: TCNT_2 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels)<br>1: TCNT_2 to TCNT_0 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible |
| 1 | SYNC1 | 0 | R/W | |
| 0 | SYNC0 | 0 | R/W | |

### 10.3.17 Timer Counter Synchronous Start Register (TCSYSTR)

TCSYSTR is an 8-bit readable/writable register that specifies synchronous start of the MTU2 and MTU2S counters. Note that the MTU2S does not have TCSYSTR.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SCH0 | SCH1 | SCH2 | SCH3 | SCH4 | - | SCH3S | SCH4S |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R | R/(W)* | R/(W)* |

Note: * Only 1 can be written to set the register.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | SCH0 | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_0 in the MTU2.<br>0: Does not specify synchronous start for TCNT_0 in the MTU2<br>1: Specifies synchronous start for TCNT_0 in the MTU2<br>[Clearing condition]<br>• When 1 is set to the CST0 bit of TSTR in MTU2 while SCH0 = 1 |
| 6 | SCH1 | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_1 in the MTU2.<br>0: Does not specify synchronous start for TCNT_1 in the MTU2<br>1: Specifies synchronous start for TCNT_1 in the MTU2<br>[Clearing condition]<br>• When 1 is set to the CST1 bit of TSTR in MTU2 while SCH1 = 1 |
| 5 | SCH2 | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_2 in the MTU2.<br>0: Does not specify synchronous start for TCNT_2 in the MTU2<br>1: Specifies synchronous start for TCNT_2 in the MTU2<br>[Clearing condition]<br>• 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<br>Controls synchronous start of TCNT_3 in the MTU2.<br>0: Does not specify synchronous start for TCNT_3 in the MTU2<br>1: Specifies synchronous start for TCNT_3 in the MTU2<br>[Clearing condition]<br>• When 1 is set to the CST3 bit of TSTR in MTU2 while SCH3 = 1 |
| 3 | SCH4 | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_4 in the MTU2.<br>0: Does not specify synchronous start for TCNT_4 in the MTU2<br>1: Specifies synchronous start for TCNT_4 in the MTU2<br>[Clearing condition]<br>• When 1 is set to the CST4 bit of TSTR in MTU2 while SCH4 = 1 |
| 2 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 1 | SCH3S | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_3S in the MTU2S.<br>0: Does not specify synchronous start for TCNT_3S in the MTU2S<br>1: Specifies synchronous start for TCNT_3S in the MTU2S<br>[Clearing condition]<br>• When 1 is set to the CST3 bit of TSTRS in MTU2S while SCH3S = 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | SCH4S | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_4S in the MTU2S.<br>0: Does not specify synchronous start for TCNT_4S in the MTU2S<br>1: Specifies synchronous start for TCNT_4S in the MTU2S<br>[Clearing condition]<br>• When 1 is set to the CST4 bit of TSTRS in MTU2S while SCH4S = 1 |

Note: * Only 1 can be written to set the register.

### 10.3.18 Timer Read/Write Enable Register (TRWER)

TRWER is an 8-bit readable/writable register that enables or disables access to the registers and counters which have write-protection capability against accidental modification in channels 3 and 4.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | RWE |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W: | R | R | R | R | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | RWE | 1 | R/W | Read/Write Enable<br>Enables or disables access to the registers which have write-protection capability against accidental modification.<br>0: Disables read/write access to the registers<br>1: Enables read/write access to the registers<br>[Clearing condition]<br>• When 0 is written to the RWE bit after reading RWE = 1 |

- Registers and counters having write-protection capability against accidental modification
  22 registers: TCR_3, TCR_4, TMDR_3, TMDR_4, TIORH_3, TIORH_4, TIORL_3, TIORL_4, TIER_3, TIER_4, TGRA_3, TGRA_4, TGRB_3, TGRB_4, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, TCNT_3, and TCNT4.

### 10.3.19 Timer Output Master Enable Register (TOER)

TOER is an 8-bit readable/writable register that enables/disables output settings for output pins TIOC4D, TIOC4C, TIOC3D, TIOC4B, TIOC4A, and TIOC3B. These pins do not output correctly if the TOER bits have not been set. Set TOER of CH3 and CH4 prior to setting TIOR of CH3 and CH4.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | OE4D | OE4C | OE3D | OE4B | OE4A | OE3B |
| Initial value: | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. |
| 5 | OE4D | 0 | R/W | Master Enable TIOC4D<br>This bit enables/disables the TIOC4D pin MTU2 output.<br>0: MTU2 output is disabled (inactive level)*<br>1: MTU2 output is enabled |
| 4 | OE4C | 0 | R/W | Master Enable TIOC4C<br>This bit enables/disables the TIOC4C pin MTU2 output.<br>0: MTU2 output is disabled (inactive level)*<br>1: MTU2 output is enabled |
| 3 | OE3D | 0 | R/W | Master Enable TIOC3D<br>This bit enables/disables the TIOC3D pin MTU2 output.<br>0: MTU2 output is disabled (inactive level)*<br>1: MTU2 output is enabled |
| 2 | OE4B | 0 | R/W | Master Enable TIOC4B<br>This bit enables/disables the TIOC4B pin MTU2 output.<br>0: MTU2 output is disabled (inactive level)*<br>1: MTU2 output is enabled |
| 1 | OE4A | 0 | R/W | Master Enable TIOC4A<br>This bit enables/disables the TIOC4A pin MTU2 output.<br>0: MTU2 output is disabled (inactive level)*<br>1: MTU2 output is enabled |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | OE3B | 0 | R/W | Master Enable TIOC3B<br>This bit enables/disables the TIOC3B pin MTU2 output.<br>0: MTU2 output is disabled (inactive level)*<br>1: MTU2 output is enabled |

Note: * The inactive level is determined by the settings in timer output control registers 1 and 2 (TOCR1 and TOCR2). For details, refer to section 10.3.20, Timer Output Control Register 1 (TOCR1), and section 10.3.21, Timer Output Control Register 2 (TOCR2). Set these bits to 1 to enable MTU2 output in other than complementary PWM or reset-synchronized PWM mode. When these bits are set to 0, low level is output.

### 10.3.20 Timer Output Control Register 1 (TOCR1)

TOCR1 is an 8-bit readable/writable register that enables/disables PWM synchronized toggle output in complementary PWM mode/reset synchronized PWM mode, and controls output level inversion of PWM output.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | PSYE | - | - | TOCL | TOCS | OLSN | OLSP |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R | R | R/(W)* | R/W | R/W | R/W |

Note: * This bit can be set to 1 only once after a power-on reset. After 1 is written, 0 cannot be written to the bit.

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PSYE | 0 | R/W | PWM Synchronous Output Enable<br>This bit selects the enable/disable of toggle output synchronized with the PWM period.<br>0: Toggle output is disabled<br>1: Toggle output is enabled |
| 5, 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | TOCL | 0 | R/(W)* | TOC Register Write Protection*[1]<br>This bit selects the enable/disable of write access to the TOCS, OLSN, and OLSP bits in TOCR1.<br>0: Write access to the TOCS, OLSN, and OLSP bits is enabled<br>1: Write access to the TOCS, OLSN, and OLSP bits is disabled |
| 2 | TOCS | 0 | R/W | TOC Select<br>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.<br>0: TOCR1 setting is selected<br>1: TOCR2 setting is selected |
| 1 | OLSN | 0 | R/W | Output Level Select N*[2]<br>This bit selects the reverse phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 10.30. |
| 0 | OLSP | 0 | R/W | Output Level Select P*[2]<br>This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 10.31. |

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

**Table 10.30 Output Level Select Function**

| Bit 1 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLSN | Initial Output | Active Level | Up Count | Down Count |
| 0 | High level | Low level | High level | Low level |
| 1 | Low level | High level | Low level | High level |

Note: The reverse phase waveform initial output value changes to active level after elapse of the dead time after count start.

**Table 10.31 Output Level Select Function**

| Bit 0 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLSP | Initial Output | Active Level | Up Count | Down Count |
| 0 | High level | Low level | Low level | High level |
| 1 | Low level | High level | High level | Low level |

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

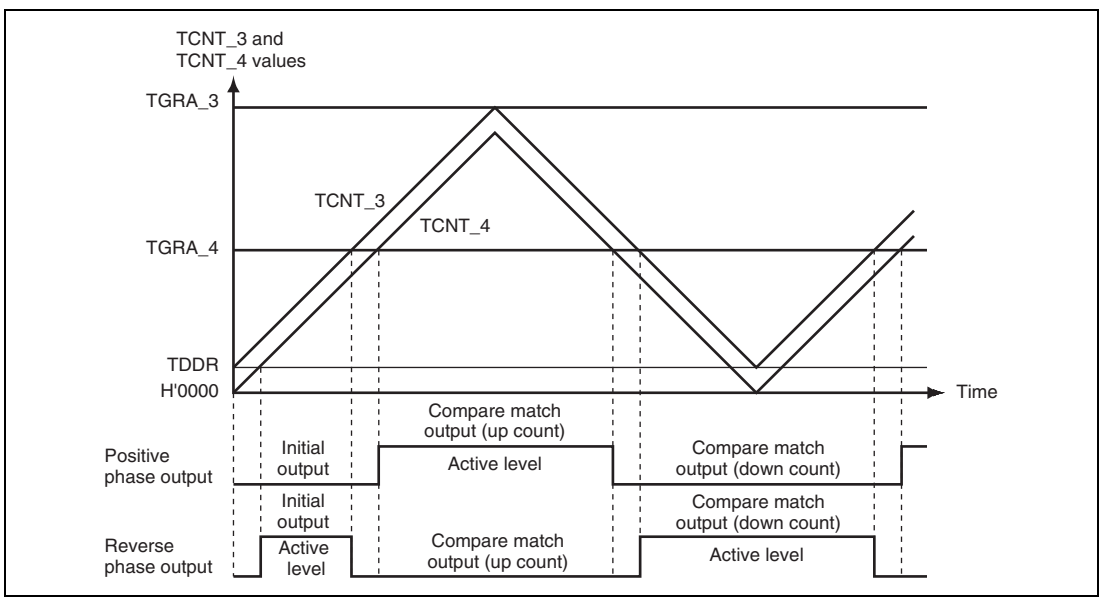



**Figure 10.2 Complementary PWM Mode Output Level Example**

### 10.3.21 Timer Output Control Register 2 (TOCR2)

TOCR2 is an 8-bit readable/writable register that controls output level inversion of PWM output in complementary PWM mode and reset-synchronized PWM mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BF[1:0] | | OLS3N | OLS3P | OLS2N | OLS2P | OLS1N | OLS1P |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | BF[1:0] | 00 | R/W | TOLBR Buffer Transfer Timing Select<br>These bits select the timing for transferring data from TOLBR to TOCR2.<br>For details, see table 10.32. |
| 5 | OLS3N | 0 | R/W | Output Level Select 3N*<br>This bit selects the output level on TIOC4D in reset-synchronized PWM mode/complementary PWM mode. See table 10.33. |
| 4 | OLS3P | 0 | R/W | Output Level Select 3P*<br>This bit selects the output level on TIOC4B in reset-synchronized PWM mode/complementary PWM mode. See table 10.34. |
| 3 | OLS2N | 0 | R/W | Output Level Select 2N*<br>This bit selects the output level on TIOC4C in reset-synchronized PWM mode/complementary PWM mode. See table 10.35. |
| 2 | OLS2P | 0 | R/W | Output Level Select 2P*<br>This bit selects the output level on TIOC4A in reset-synchronized PWM mode/complementary PWM mode. See table 10.36. |
| 1 | OLS1N | 0 | R/W | Output Level Select 1N*<br>This bit selects the output level on TIOC3D in reset-synchronized PWM mode/complementary PWM mode. See table 10.37. |
| 0 | OLS1P | 0 | R/W | Output Level Select 1P*<br>This bit selects the output level on TIOC3B in reset-synchronized PWM mode/complementary PWM mode. See table 10.38. |

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

**Table 10.32 Setting of Bits BF1 and BF0**

| Bit 7 | Bit 6 | Description | |
|---|---|---|---|
| **BF1** | **BF0** | **Complementary PWM Mode** | **Reset-Synchronized PWM Mode** |
| 0 | 0 | Does not transfer data from the buffer register (TOLBR) to TOCR2. | Does not transfer data from the buffer register (TOLBR) to TOCR2. |
| 0 | 1 | Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the TCNT_4 count. | Transfers data from the buffer register (TOLBR) to TOCR2 when TCNT_3/TCNT_4 is cleared |
| 1 | 0 | Transfers data from the buffer register (TOLBR) to TOCR2 at the trough of the TCNT_4 count. | Setting prohibited |
| 1 | 1 | Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the TCNT_4 count. | Setting prohibited |

**Table 10.33 TIOC4D Output Level Select Function**

| Bit 5 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| **OLS3N** | **Initial Output** | **Active Level** | **Up Count** | **Down Count** |
| 0 | High level | Low level | High level | Low level |
| 1 | Low level | High level | Low level | High level |

Note: The reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

**Table 10.34 TIOC4B Output Level Select Function**

| Bit 4 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| **OLS3P** | **Initial Output** | **Active Level** | **Up Count** | **Down Count** |
| 0 | High level | Low level | Low level | High level |
| 1 | Low level | High level | High level | Low level |

**Table 10.35 TIOC4C Output Level Select Function**

| Bit 3 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS2N | Initial Output | Active Level | Up Count | Down Count |
| 0 | High level | Low level | High level | Low level |
| 1 | Low level | High level | Low level | High level |

Note: The reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

**Table 10.36 TIOC4A Output Level Select Function**

| Bit 2 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS2P | Initial Output | Active Level | Up Count | Down Count |
| 0 | High level | Low level | Low level | High level |
| 1 | Low level | High level | High level | Low level |

**Table 10.37 TIOC3D Output Level Select Function**

| Bit 1 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS1N | Initial Output | Active Level | Up Count | Down Count |
| 0 | High level | Low level | High level | Low level |
| 1 | Low level | High level | Low level | High level |

Note: The reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

**Table 10.38 TIOC4B Output Level Select Function**

| Bit 0 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS1P | Initial Output | Active Level | Up Count | Down Count |
| 0 | High level | Low level | Low level | High level |
| 1 | Low level | High level | High level | Low level |

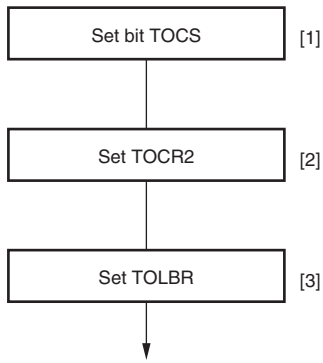
### 10.3.22 Timer Output Level Buffer Register (TOLBR)

TOLBR is an 8-bit readable/writable register that functions as a buffer for TOCR2 and specifies the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | OLS3N | OLS3P | OLS2N | OLS2P | OLS1N | OLS1P |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5 | OLS3N | 0 | R/W | Specifies the buffer value to be transferred to the OLS3N bit in TOCR2. |
| 4 | OLS3P | 0 | R/W | Specifies the buffer value to be transferred to the OLS3P bit in TOCR2. |
| 3 | OLS2N | 0 | R/W | Specifies the buffer value to be transferred to the OLS2N bit in TOCR2. |
| 2 | OLS2P | 0 | R/W | Specifies the buffer value to be transferred to the OLS2P bit in TOCR2. |
| 1 | OLS1N | 0 | R/W | Specifies the buffer value to be transferred to the OLS1N bit in TOCR2. |
| 0 | OLS1P | 0 | R/W | Specifies the buffer value to be transferred to the OLS1P bit in TOCR2. |

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




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

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

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

**Figure 10.3 PWM Output Level Setting Procedure in Buffer Operation**

### 10.3.23 Timer Gate Control Register (TGCR)

TGCR is an 8-bit readable/writable register that controls the waveform output necessary for brushless DC motor control in reset-synchronized PWM mode/complementary PWM mode. These register settings are ineffective for anything other than complementary PWM mode/reset-synchronized PWM mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | BDC | N | P | FB* | WF | VF | UF |
| Initial value: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 6 | BDC | 0 | R/W | Brushless DC Motor<br>This bit selects whether to make the functions of this register (TGCR) effective or ineffective.<br>0: Ordinary output<br>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<br>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.<br>0: Level output<br>1: Reset synchronized PWM/complementary PWM output |
| 4 | P | 0 | R/W | Positive Phase Output (P) Control<br>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.<br>0: Level output<br>1: Reset synchronized PWM/complementary PWM output |
| 3 | FB* | 0 | R/W | External Feedback Signal Enable<br>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.<br>0: Output switching is external input (Input sources are channel 0 TGRA, TGRB, TGRC input capture signal)<br>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<br>These bits set the positive phase/negative phase output phase on or off state. The setting of these bits is valid only when the FB bit in this register is set to 1. In this case, the setting of bits 2 to 0 is a substitute for external input. See table 10.39. |
| 1 | VF | 0 | R/W | |
| 0 | UF | 0 | R/W | |

Note: * When the MTU2S is used to set the BDC bit to 1, do not set the FB bit to 0.

**Table 10.39 Output level Select Function**

| | | | Function | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bit 2** | **Bit 1** | **Bit 0** | **TIOC3B** | **TIOC4A** | **TIOC4B** | **TIOC3D** | **TIOC4C** | **TIOC4D** |
| **WF** | **VF** | **UF** | **U Phase** | **V Phase** | **W Phase** | **U Phase** | **V Phase** | **W Phase** |
| 0 | 0 | 0 | OFF | OFF | OFF | OFF | OFF | OFF |
| | | 1 | ON | OFF | OFF | OFF | OFF | ON |
| | 1 | 0 | OFF | ON | OFF | ON | OFF | OFF |
| | | 1 | OFF | ON | OFF | OFF | OFF | ON |
| 1 | 0 | 0 | OFF | OFF | ON | OFF | ON | OFF |
| | | 1 | ON | OFF | OFF | OFF | ON | OFF |
| | 1 | 0 | OFF | OFF | ON | ON | OFF | OFF |
| | | 1 | OFF | OFF | OFF | OFF | OFF | OFF |

### 10.3.24 Timer Subcounter (TCNTS)

TCNTS is a 16-bit read-only counter that is used only in complementary PWM mode.

The initial value of TCNTS is H'0000.

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

Note: Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

### 10.3.25 Timer Dead Time Data Register (TDDR)

TDDR is a 16-bit register, used only in complementary PWM mode, that specifies the TCNT_3 and TCNT_4 counter offset values. In complementary PWM mode, when the TCNT_3 and TCNT_4 counters are cleared and then restarted, the TDDR register value is loaded into the TCNT_3 counter and the count operation starts.

The initial value of TDDR is H'FFFF.

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

Note: Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

### 10.3.26 Timer Cycle Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value as the TCDR register value. This register is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (decrement to increment).

The initial value of TCDR is H'FFFF.

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

Note: Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

### 10.3.27 Timer Cycle Buffer Register (TCBR)

TCBR is a 16-bit register used only in complementary PWM mode. It functions as a buffer register for the TCDR register. The TCBR register values are transferred to the TCDR register with the transfer timing set in the TMDR register.

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.

### 10.3.28 Timer Interrupt Skipping Set Register (TITCR)

TITCR is an 8-bit readable/writable register that enables or disables interrupt skipping and specifies the interrupt skipping count. The MTU2 has one TITCR.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | T3AEN | 3ACOR[2:0] | | | T4VEN | 4VCOR[2:0] | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 | T3AEN | 0 | R/W | T3AEN<br>Enables or disables TGIA_3 interrupt skipping.<br>0: TGIA_3 interrupt skipping disabled<br>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.*<br>For details, see table 10.40. |
| 3 | T4VEN | 0 | R/W | T4VEN<br>Enables or disables TCIV_4 interrupt skipping.<br>0: TCIV_4 interrupt skipping disabled<br>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.*<br>For details, see table 10.41. |

Note: * When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed. Before changing the interrupt skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter (TITCNT).

**Table 10.40 Setting of Interrupt Skipping Count by Bits 3ACOR2 to 3ACOR0**

| Bit 6 | Bit 5 | Bit 4 | |
|---|---|---|---|
| **3ACOR2** | **3ACOR1** | **3ACOR0** | **Description** |
| 0 | 0 | 0 | Does not skip TGIA_3 interrupts. |
| 0 | 0 | 1 | Sets the TGIA_3 interrupt skipping count to 1. |
| 0 | 1 | 0 | Sets the TGIA_3 interrupt skipping count to 2. |
| 0 | 1 | 1 | Sets the TGIA_3 interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TGIA_3 interrupt skipping count to 4. |
| 1 | 0 | 1 | Sets the TGIA_3 interrupt skipping count to 5. |
| 1 | 1 | 0 | Sets the TGIA_3 interrupt skipping count to 6. |
| 1 | 1 | 1 | Sets the TGIA_3 interrupt skipping count to 7. |

**Table 10.41 Setting of Interrupt Skipping Count by Bits 4VCOR2 to 4VCOR0**

| Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|
| **4VCOR2** | **4VCOR1** | **4VCOR0** | **Description** |
| 0 | 0 | 0 | Does not skip TCIV_4 interrupts. |
| 0 | 0 | 1 | Sets the TCIV_4 interrupt skipping count to 1. |
| 0 | 1 | 0 | Sets the TCIV_4 interrupt skipping count to 2. |
| 0 | 1 | 1 | Sets the TCIV_4 interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TCIV_4 interrupt skipping count to 4. |
| 1 | 0 | 1 | Sets the TCIV_4 interrupt skipping count to 5. |
| 1 | 1 | 0 | Sets the TCIV_4 interrupt skipping count to 6. |
| 1 | 1 | 1 | Sets the TCIV_4 interrupt skipping count to 7. |

### 10.3.29 Timer Interrupt Skipping Counter (TITCNT)

TITCNT is an 8-bit readable/writable counter. The MTU2 has one TITCNT. TITCNT retains its value even after stopping the count operation of TCNT_3 and TCNT_4.

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. |
| 6 to 4 | 3ACNT[2:0] | 000 | R | TGIA_3 Interrupt Counter<br>While the T3AEN bit in TITCR is set to 1, the count in these bits is incremented every time a TGIA_3 interrupt occurs.<br>[Clearing conditions]<br>• When the 3ACNT2 to 3ACNT0 value in TITCNT matches the 3ACOR2 to 3ACOR0 value in TITCR<br>• When the T3AEN bit in TITCR is cleared to 0<br>• When the 3ACOR2 to 3ACOR0 bits in TITCR are cleared to 0 |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. |
| 2 to 0 | 4VCNT[2:0] | 000 | R | TCIV_4 Interrupt Counter<br>While the T4VEN bit in TITCR is set to 1, the count in these bits is incremented every time a TCIV_4 interrupt occurs.<br>[Clearing conditions]<br>• When the 4VCNT2 to 4VCNT0 value in TITCNT matches the 4VCOR2 to 4VCOR2 value in TITCR<br>• When the T4VEN bit in TITCR is cleared to 0<br>• When the 4VCOR2 to 4VCOR2 bits in TITCR are cleared to 0 |

Note: To clear the TITCNT, clear the T3AEN and T4VEN bits in TITCR to 0.

### 10.3.30 Timer Buffer Transfer Set Register (TBTER)

TBTER is an 8-bit readable/writable register that enables or disables transfer from the buffer registers* used in complementary PWM mode to the temporary registers and specifies whether to link the transfer with interrupt skipping operation. The MTU2 has one TBTER.

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 2 | — | All 0 | R | Reserved<br>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.<br>For details, see table 10.42. |

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

**Table 10.42 Setting of Bits BTE1 and BTE0**

| Bit 1 | Bit 0 | |
|---|---|---|
| **BTE1** | **BTE0** | **Description** |
| 0 | 0 | Enables transfer from the buffer registers to the temporary registers*[1] and does not link the transfer with interrupt skipping operation. |
| 0 | 1 | Disables transfer from the buffer registers to the temporary registers. |
| 1 | 0 | Links transfer from the buffer registers to the temporary registers with interrupt skipping operation.*[2] |
| 1 | 1 | Setting prohibited |

Notes: 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 10.4.8, Complementary PWM Mode.
2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0)), be sure to disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

### 10.3.31 Timer Dead Time Enable Register (TDER)

TDER is an 8-bit readable/writable register that controls dead time generation in complementary PWM mode. The MTU2 has one TDER in channel 3. TDER must be modified only while TCNT stops.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | TDER |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W: | R | R | R | R | R | R | R | R/(W) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | TDER | 1 | R/(W) | Dead Time Enable<br>Specifies whether to generate dead time.<br>0: Does not generate dead time<br>1: Generates dead time*<br>[Clearing condition]<br>• When 0 is written to TDER after reading TDER = 1 |

Note: * TDDR must be set to 1 or a larger value.

### 10.3.32 Timer Waveform Control Register (TWCR)

TWCR is an 8-bit readable/writable register that controls the waveform when synchronous counter clearing occurs in TCNT_3 and TCNT_4 in complementary PWM mode and specifies whether to clear the counters at TGRA_3 compare match. The CCE bit and WRE bit in TWCR must be modified only while TCNT stops.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCE | - | - | - | - | - | SCC | WRE |
| Initial value: | 0* | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/(W) | R | R | R | R | R | R/(W) | R/(W) |

Note: * Do not set to 1 when complementary PWM mode is not selected.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CCE | 0* | R/(W) | Compare Match Clear Enable<br>Specifies whether to clear counters at TGRA_3 compare match in complementary PWM mode.<br>0: Does not clear counters at TGRA_3 compare match<br>1: Clears counters at TGRA_3 compare match<br>[Setting condition]<br>• When 1 is written to CCE after reading CCE = 0 |
| 6 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | SCC | 0 | R/(W) | Synchronous Clearing Control<br>Specifies whether to clear TCNT_3 and TCNT_4 in the MTU2S when synchronous counter clearing between the MTU2 and MTU2S occurs in complementary PWM mode.<br>When using this control, place the MTU2S in complementary PWM mode.<br>When modifying the SCC bit while the counters are operating, do not modify the CCE or WRE bits.<br>Counter clearing synchronized with the MTU2 is disabled by the SCC bit setting only when synchronous clearing occurs outside the Tb interval at the trough. When synchronous clearing occurs in the Tb interval at the trough including the period immediately after TCNT_3 and TCNT_4 start operation, TCNT_3 and TCNT_4 in the MTU2S are cleared.<br>For the Tb interval at the trough in complementary PWM mode, see figure 10.40.<br>In the MTU2, this bit is reserved. It is always read as 0 and the write value should always be 0.<br>0: Enables clearing of TCNT_3 and TCNT_4 in the MTU2S by MTU2–MTU2S synchronous clearing operation<br>1: Disables clearing of TCNT_3 and TCNT_4 in the MTU2S by MTU2–MTU2S synchronous clearing operation<br>[Setting condition]<br>• When 1 is written to SCC after reading SCC = 0 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | WRE | 0 | R/(W) | Initial Output Suppression Enable<br>Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.<br>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.<br>For the Tb interval at the trough in complementary PWM mode, see figure 10.40.<br>0: Outputs the initial value specified in TOCR<br>1: Suppresses initial output<br>[Setting condition]<br>• When 1 is written to WRE after reading WRE = 0 |

Note: * Do not set to 1 when complementary PWM mode is not selected.

### 10.3.33 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCOR), and timer A/D converter start request cycle set buffer registers (TADCOBR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/writes. 8-bit read/write is not possible. Always access in 16-bit units.

All registers other than the above registers are 8-bit registers. These are connected to the CPU by a 16-bit data bus, so 16-bit read/writes and 8-bit read/writes are both possible.

## 10.4 Operation

### 10.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, cycle counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Always select MTU2 external pins set function using the pin function controller (PFC).

**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.

1. Example of Count Operation Setting Procedure
   Figure 10.4 shows an example of the count operation setting procedure.

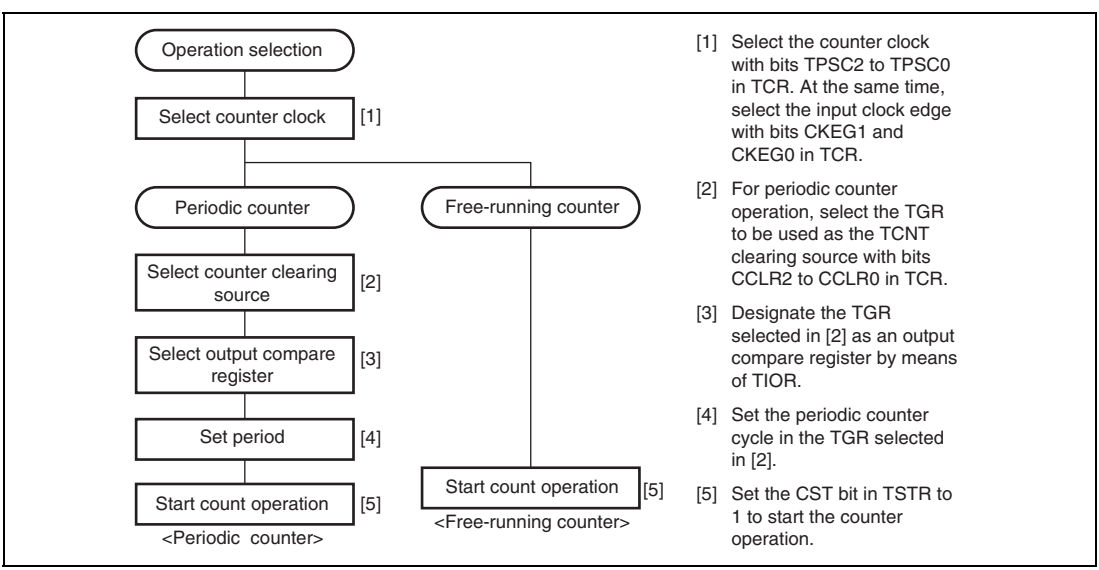



**Figure 10.4 Example of Counter Operation Setting Procedure**

2. Free-Running Count Operation and Periodic Count Operation:

   Immediately after a reset, the MTU2's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the MTU2 requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

   Figure 10.5 illustrates free-running counter operation.

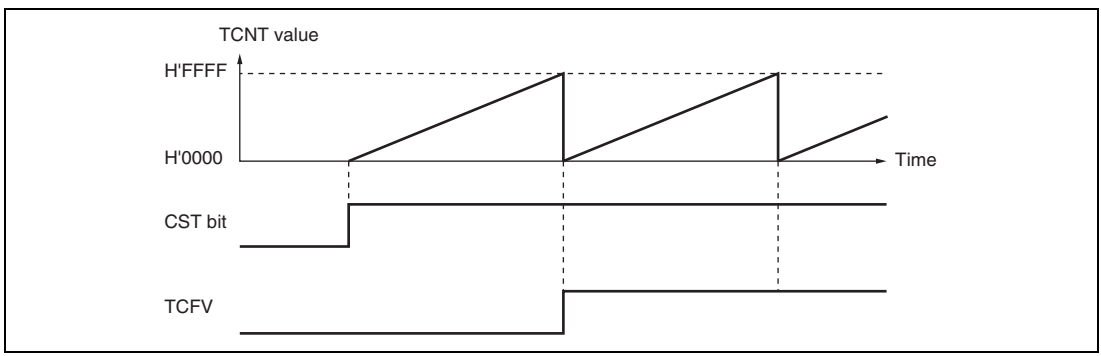



**Figure 10.5 Free-Running Counter Operation**

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR0 to CCLR2 in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the MTU2 requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 10.6 illustrates periodic counter operation.

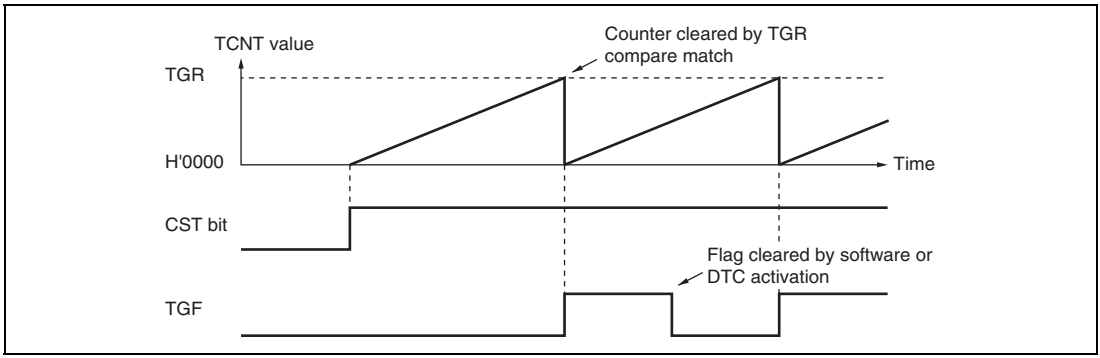



**Figure 10.6 Periodic Counter Operation**

**Waveform Output by Compare Match:**

The MTU2 can perform 0, 1, or toggle output from the corresponding output pin using compare match.

1. Example of Setting Procedure for Waveform Output by Compare Match
   Figure 10.7 shows an example of the setting procedure for waveform output by compare match

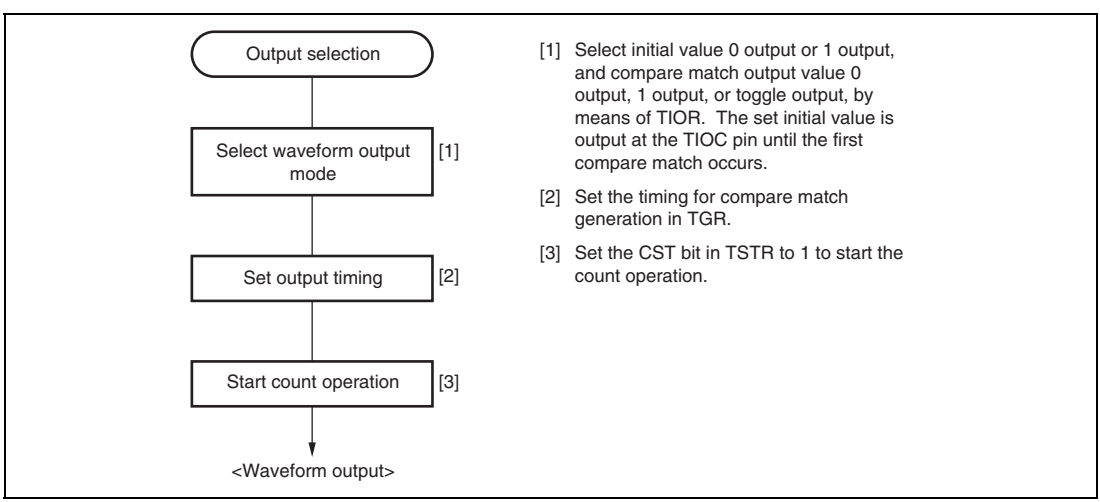



**Figure 10.7 Example of Setting Procedure for Waveform Output by Compare Match**

2. Examples of Waveform Output Operation:

   Figure 10.8 shows an example of 0 output/1 output.

   In this example TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

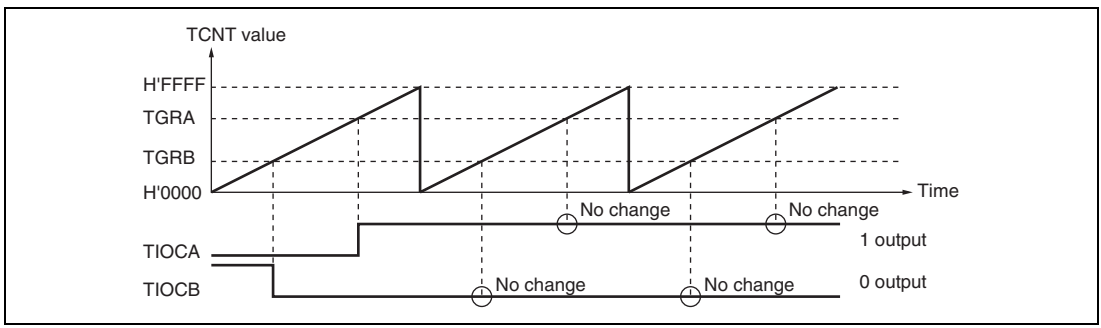



**Figure 10.8 Example of 0 Output/1 Output Operation**

Figure 10.9 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

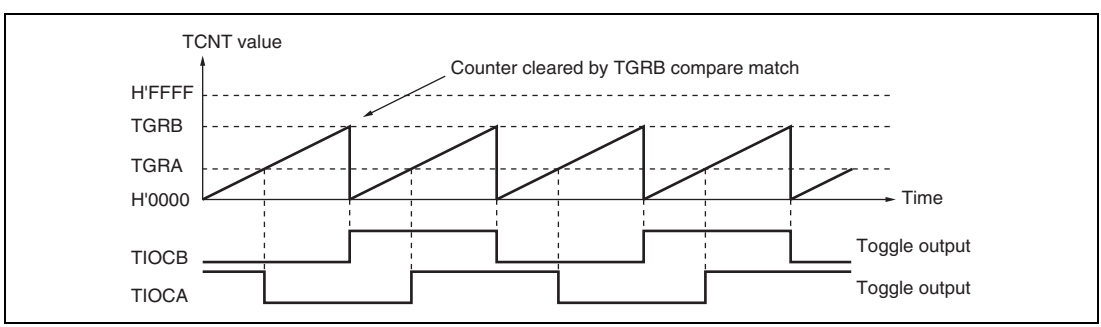



**Figure 10.9 Example of Toggle Output Operation**

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

1. Example of Input Capture Operation Setting Procedure

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

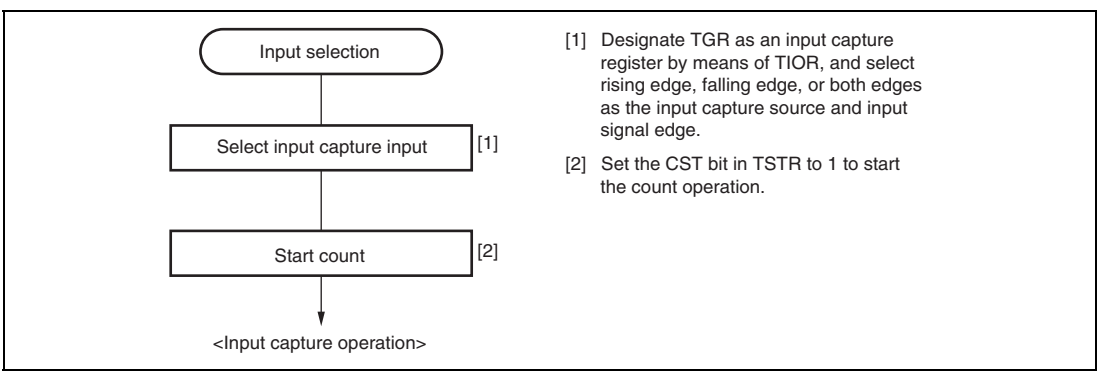



**Figure 10.10 Example of Input Capture Operation Setting Procedure**

2. Example of Input Capture Operation:

   Figure 10.11 shows an example of input capture operation.

   In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, the falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

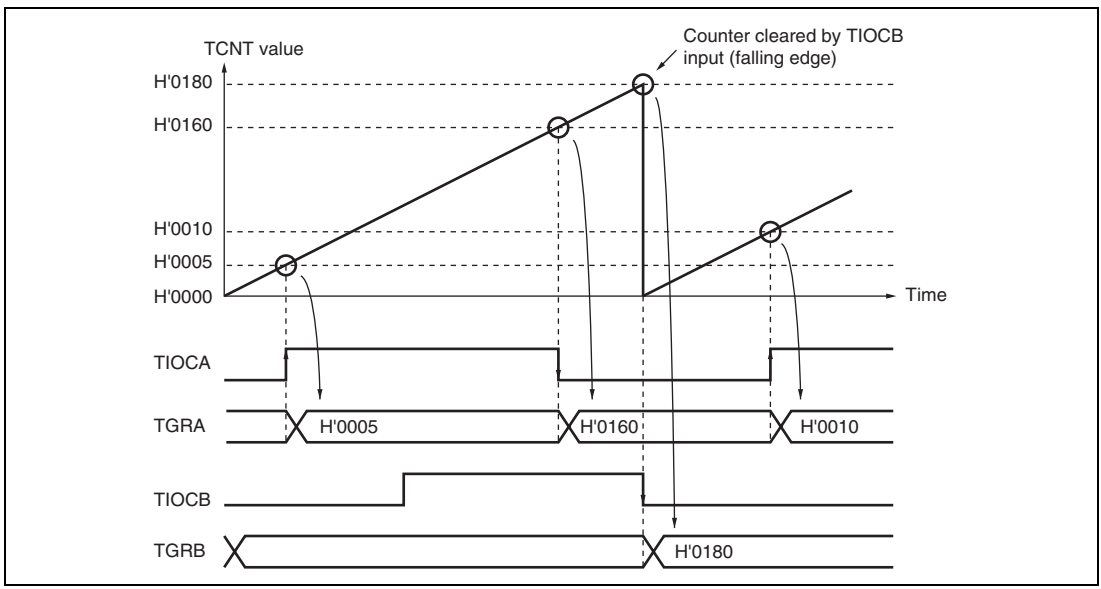



**Figure 10.11 Example of Input Capture Operation**

### 10.4.2 Synchronous Operation

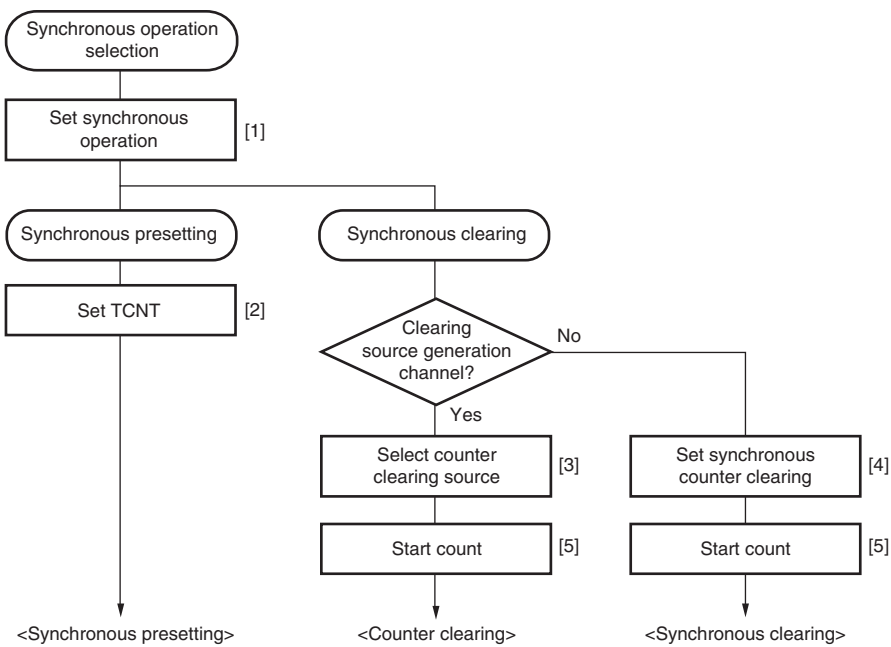
In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 4 can all be designated for synchronous operation. Channel 5 cannot be used for synchronous operation.

**Example of Synchronous Operation Setting Procedure:**

Figure 10.12 shows an example of the synchronous operation setting procedure.




[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

**Figure 10.12 Example of Synchronous Operation Setting Procedure**

**Example of Synchronous Operation:** Figure 10.13 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, are performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details of PWM modes, see section 10.4.5, PWM Modes.

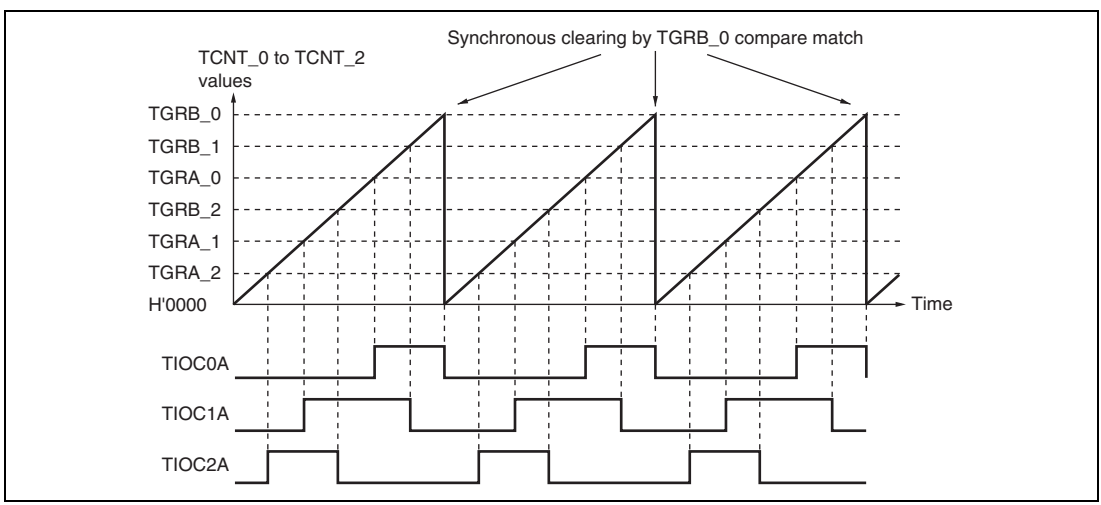



**Figure 10.13 Example of Synchronous Operation**

### 10.4.3 Buffer Operation

Buffer operation, provided for channels 0, 3, and 4, enables TGRC and TGRD to be used as buffer registers. In channel 0, TGRF can also be used as a buffer register.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: TGRE_0 cannot be designated as an input capture register and can only operate as a compare match register.

Table 10.43 shows the register combinations used in buffer operation.

**Table 10.43 Register Combinations in Buffer Operation**

| Channel | Timer General Register | Buffer Register |
|---|---|---|
| 0 | TGRA_0 | TGRC_0 |
| | TGRB_0 | TGRD_0 |
| | TGRE_0 | TGRF_0 |
| 3 | TGRA_3 | TGRC_3 |
| | TGRB_3 | TGRD_3 |
| 4 | TGRA_4 | TGRC_4 |
| | TGRB_4 | TGRD_4 |

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

  This operation is illustrated in figure 10.14.

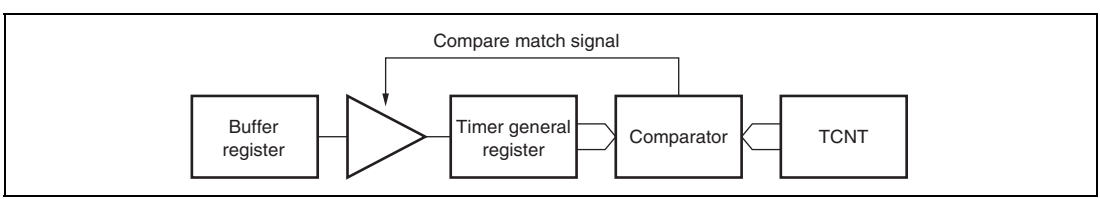



**Figure 10.14 Compare Match Buffer Operation**

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 10.15.

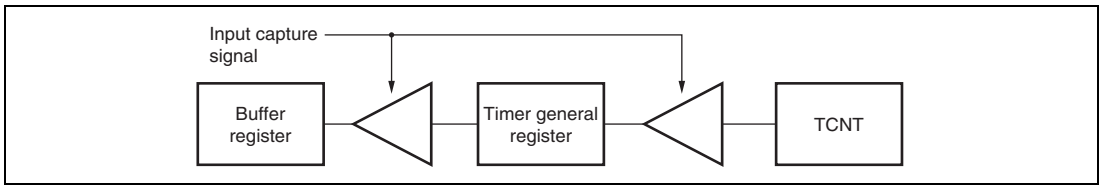



**Figure 10.15 Input Capture Buffer Operation**

**Example of Buffer Operation Setting Procedure:** Figure 10.16 shows an example of the buffer operation setting procedure.

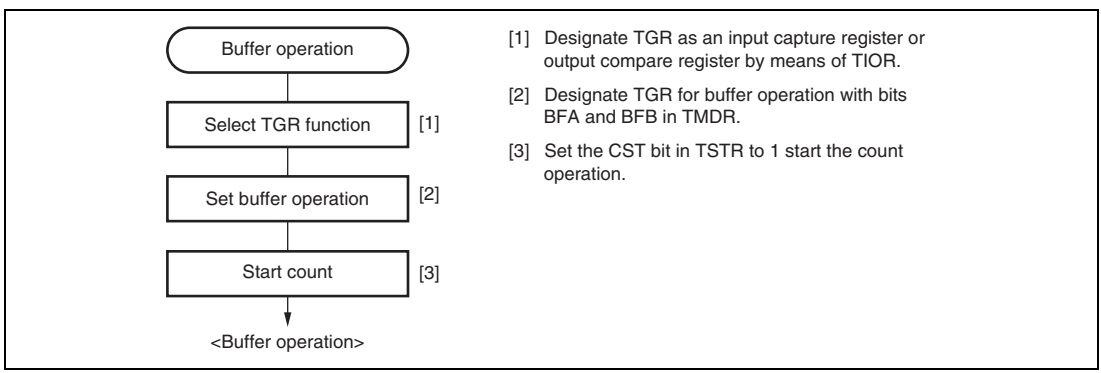



**Figure 10.16 Example of Buffer Operation Setting Procedure**

**Examples of Buffer Operation:**

1. When TGR is an output compare register

   Figure 10.17 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. In this example, the TTSA bit in TBTM is cleared to 0.

   As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time that compare match A occurs.

   For details of PWM modes, see section 10.4.5, PWM Modes.

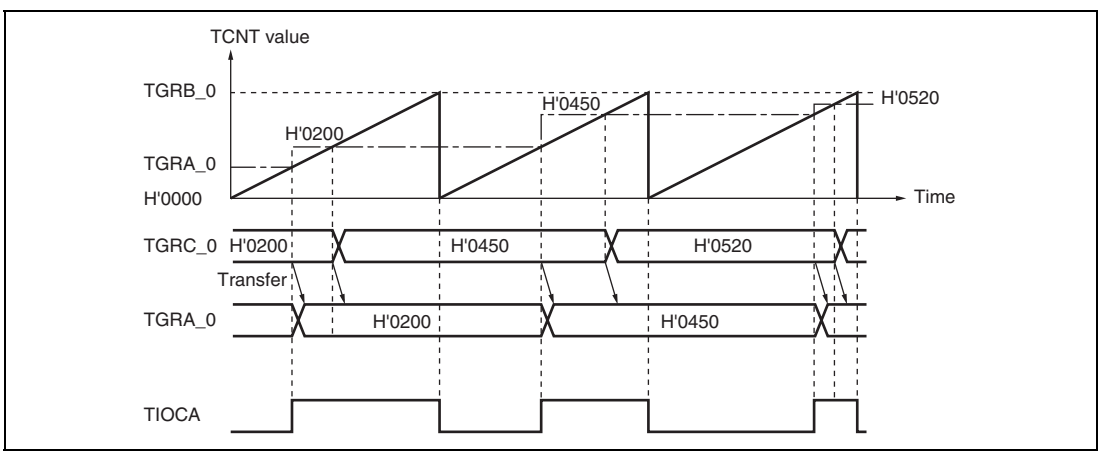



**Figure 10.17 Example of Buffer Operation (1)**

2. When TGR is an input capture register

   Figure 10.18 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

   Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

   As buffer operation has been set, when the TCNT value is stored in TGRA upon the occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

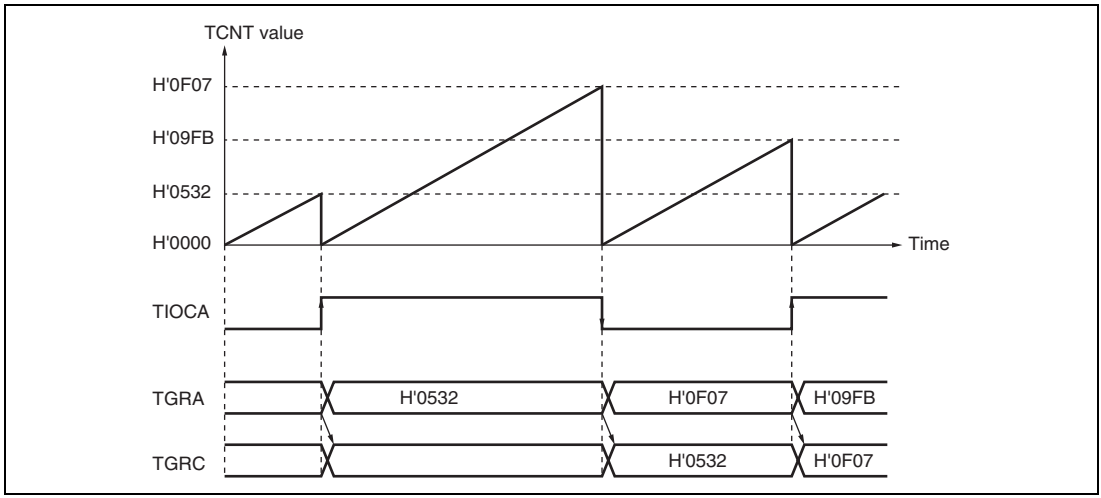



**Figure 10.18 Example of Buffer Operation (2)**

**Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation:** The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for channel 0 or in PWM mode 1 for channels 3 and 4 by setting the buffer operation transfer mode registers (TBTM_0, TBTM_3, and TBTM_4). Either compare match (initial setting) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (H'FFFF to H'0000)
- When H'0000 is written to TCNT during counting
- When TCNT is cleared to H'0000 under the condition specified in the CCLR2 to CCLR0 bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 10.19 shows an operation example in which PWM mode 1 is designated for channel 0 and buffer operation is designated for TGRA_0 and TGRC_0. The settings used in this example are TCNT_0 clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. The TTSA bit in TBTM_0 is set to 1.

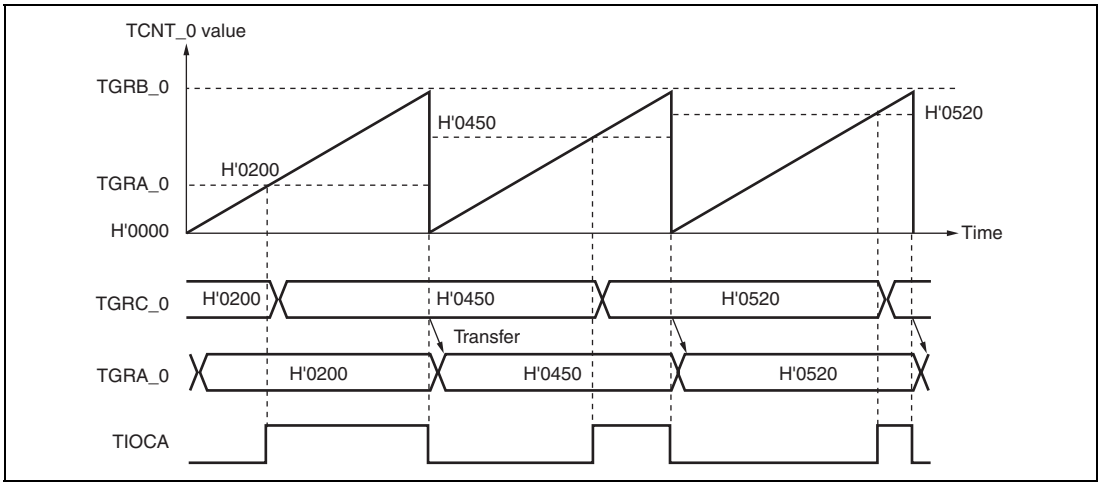



**Figure 10.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing**

### 10.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 counter clock upon overflow/underflow of TCNT_2 as set in bits TPSC0 to TPSC2 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 10.44 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1, the counter clock setting is invalid and the counters operates independently in phase counting mode.

**Table 10.44 Cascaded Combinations**

| Combination | Upper 16 Bits | Lower 16 Bits |
|---|---|---|
| Channels 1 and 2 | TCNT_1 | TCNT_2 |

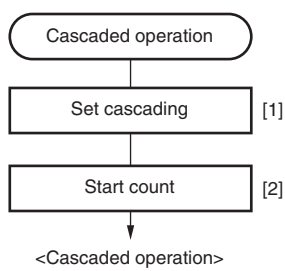
For simultaneous input capture of TCNT_1 and TCNT_2 during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). For input capture in cascade connection, refer to section 10.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.

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

**Table 10.45 TICCR Setting and Input Capture Input Pins**

| Target Input Capture | TICCR Setting | Input Capture Input Pins |
|---|---|---|
| Input capture from TCNT_1 to TGRA_1 | I2AE bit = 0 (initial value) | TIOC1A |
| | I2AE bit = 1 | TIOC1A, TIOC2A |
| Input capture from TCNT_1 to TGRB_1 | I2BE bit = 0 (initial value) | TIOC1B |
| | I2BE bit = 1 | TIOC1B, TIOC2B |
| Input capture from TCNT_2 to TGRA_2 | I1AE bit = 0 (initial value) | TIOC2A |
| | I1AE bit = 1 | TIOC2A, TIOC1A |
| Input capture from TCNT_2 to TGRB_2 | I1BE bit = 0 (initial value) | TIOC2B |
| | I1BE bit = 1 | TIOC2B, TIOC1B |

**Example of Cascaded Operation Setting Procedure:** Figure 10.20 shows an example of the setting procedure for cascaded operation.




[1] Set bits TPSC2 to TPSC0 in the channel 1 TCR to B'1111 to select TCNT_2 overflow/ underflow counting.

[2] Set the CST bit in TSTR for the upper and lower channel to 1 to start the count operation.

**Figure 10.20 Cascaded Operation Setting Procedure**

**Cascaded Operation Example (a):** Figure 10.21 illustrates the operation when TCNT_2 overflow/underflow counting has been set for TCNT_1 and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

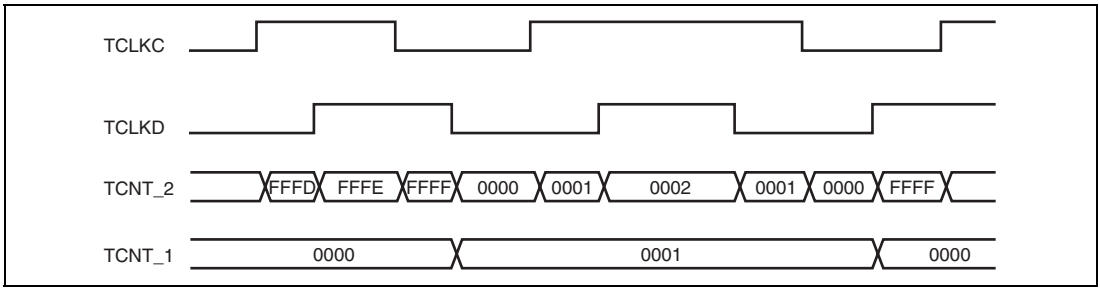



**Figure 10.21 Cascaded Operation Example (a)**

**Cascaded Operation Example (b):** Figure 10.22 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected the TIOC1A rising edge for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both TIOC1A and TIOC2A is used for the TGRA_1 input capture condition. For the TGRA_2 input capture condition, the TIOC2A rising edge is used.

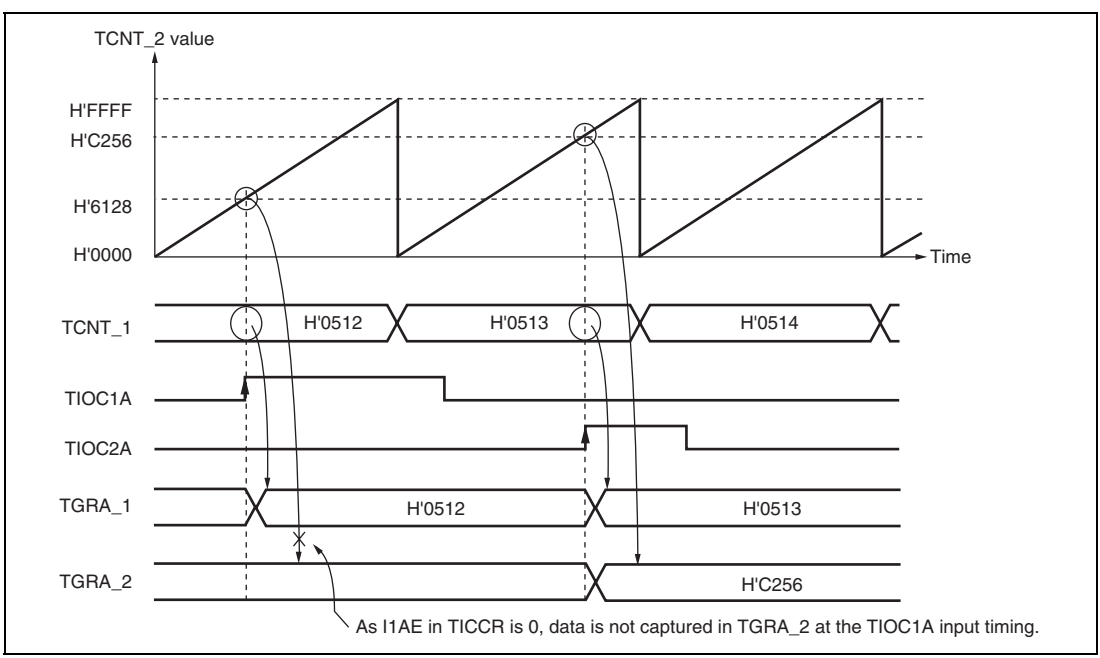



**Figure 10.22 Cascaded Operation Example (b)**

**Cascaded Operation Example (c):** Figure 10.23 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE and I1AE bits in TICCR have been set to 1 to include the TIOC2A and TIOC1A pins in the TGRA_1 and TGRA_2 input capture conditions, respectively. In this example, the IOA0 to IOA3 bits in both TIOR_1 and TIOR_2 have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of TIOC1A and TIOC2A input is used for the TGRA_1 and TGRA_2 input capture conditions.

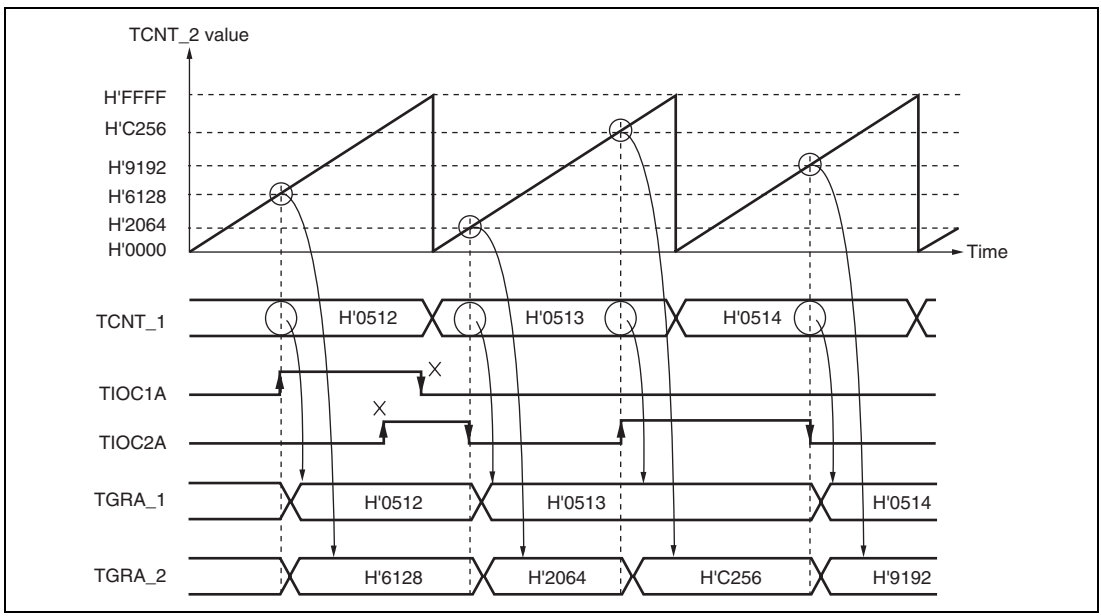



**Figure 10.23 Cascaded Operation Example (c)**

**Cascaded Operation Example (d):** Figure 10.24 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected TGRA_0 compare match or input capture occurrence for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, as TIOR_1 has selected TGRA_0 compare match or input capture occurrence for the input capture timing, the TIOC2A edge is not used for TGRA_1 input capture condition although the I2AE bit in TICCR has been set to 1.

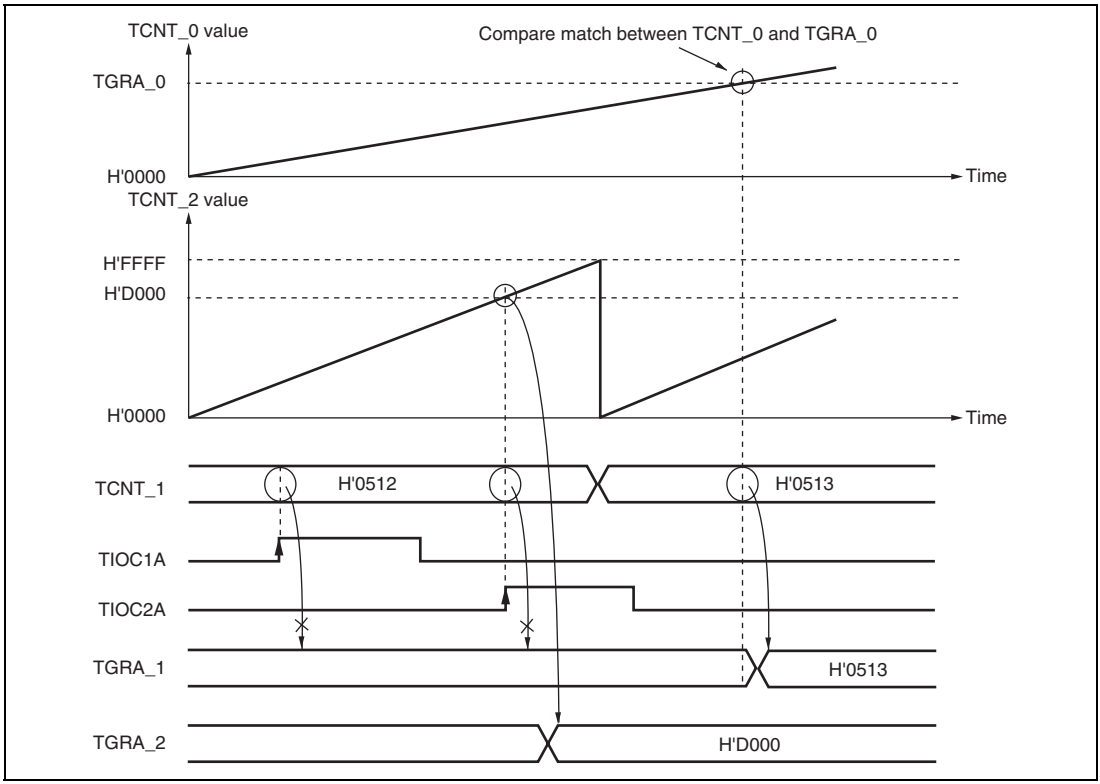



**Figure 10.24 Cascaded Operation Example (d)**

### 10.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can be selected as 0, 1, or toggle output in response to a compare match of each TGR.

TGR registers settings can be used to output a PWM waveform in the range of 0% to 100% duty.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

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 10.46.

**Table 10.46 PWM Output Registers and Output Pins**

| | | Output Pins | |
|---|---|---|---|
| **Channel** | **Registers** | **PWM Mode 1** | **PWM Mode 2** |
| 0 | TGRA_0 | TIOC0A | TIOC0A |
| | TGRB_0 | | TIOC0B |
| | TGRC_0 | TIOC0C | TIOC0C |
| | TGRD_0 | | TIOC0D |
| 1 | TGRA_1 | TIOC1A | TIOC1A |
| | TGRB_1 | | TIOC1B |
| 2 | TGRA_2 | TIOC2A | TIOC2A |
| | TGRB_2 | | TIOC2B |
| 3 | TGRA_3 | TIOC3A | Cannot be set |
| | TGRB_3 | | Cannot be set |
| | TGRC_3 | TIOC3C | Cannot be set |
| | TGRD_3 | | Cannot be set |
| 4 | TGRA_4 | TIOC4A | Cannot be set |
| | TGRB_4 | | Cannot be set |
| | TGRC_4 | TIOC4C | Cannot be set |
| | TGRD_4 | | Cannot be set |

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

**Example of PWM Mode Setting Procedure:** Figure 10.25 shows an example of the PWM mode setting procedure.

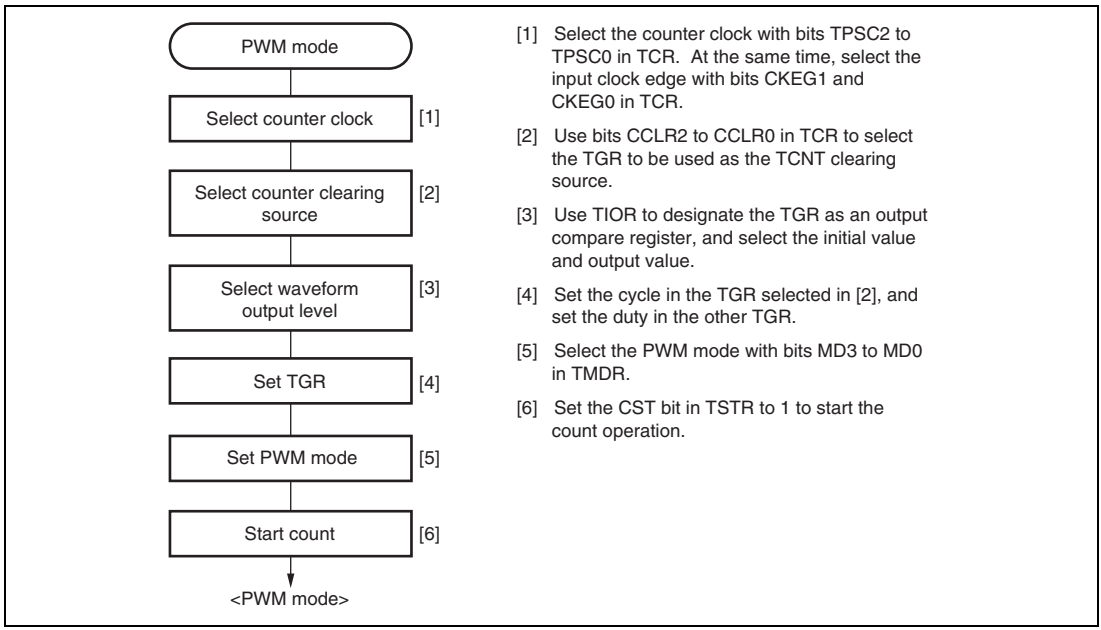



**Figure 10.25 Example of PWM Mode Setting Procedure**

**Examples of PWM Mode Operation:** Figure 10.26 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB registers are used as the duty levels.

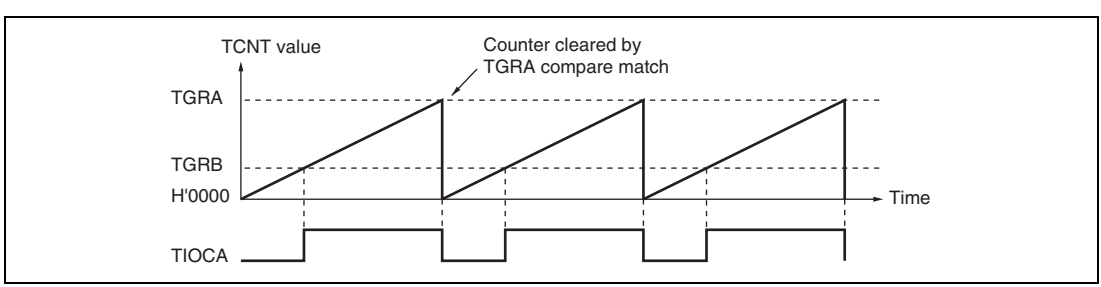



**Figure 10.26 Example of PWM Mode Operation (1)**

Figure 10.27 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), outputting a 5-phase PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs are used as the duty levels.

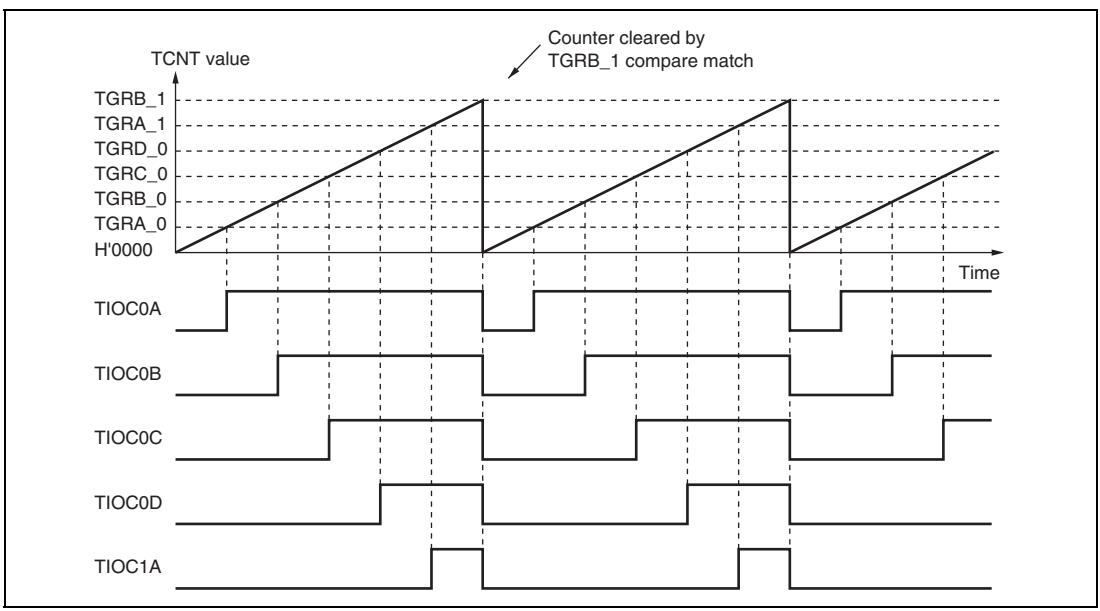



**Figure 10.27 Example of PWM Mode Operation (2)**

Figure 10.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

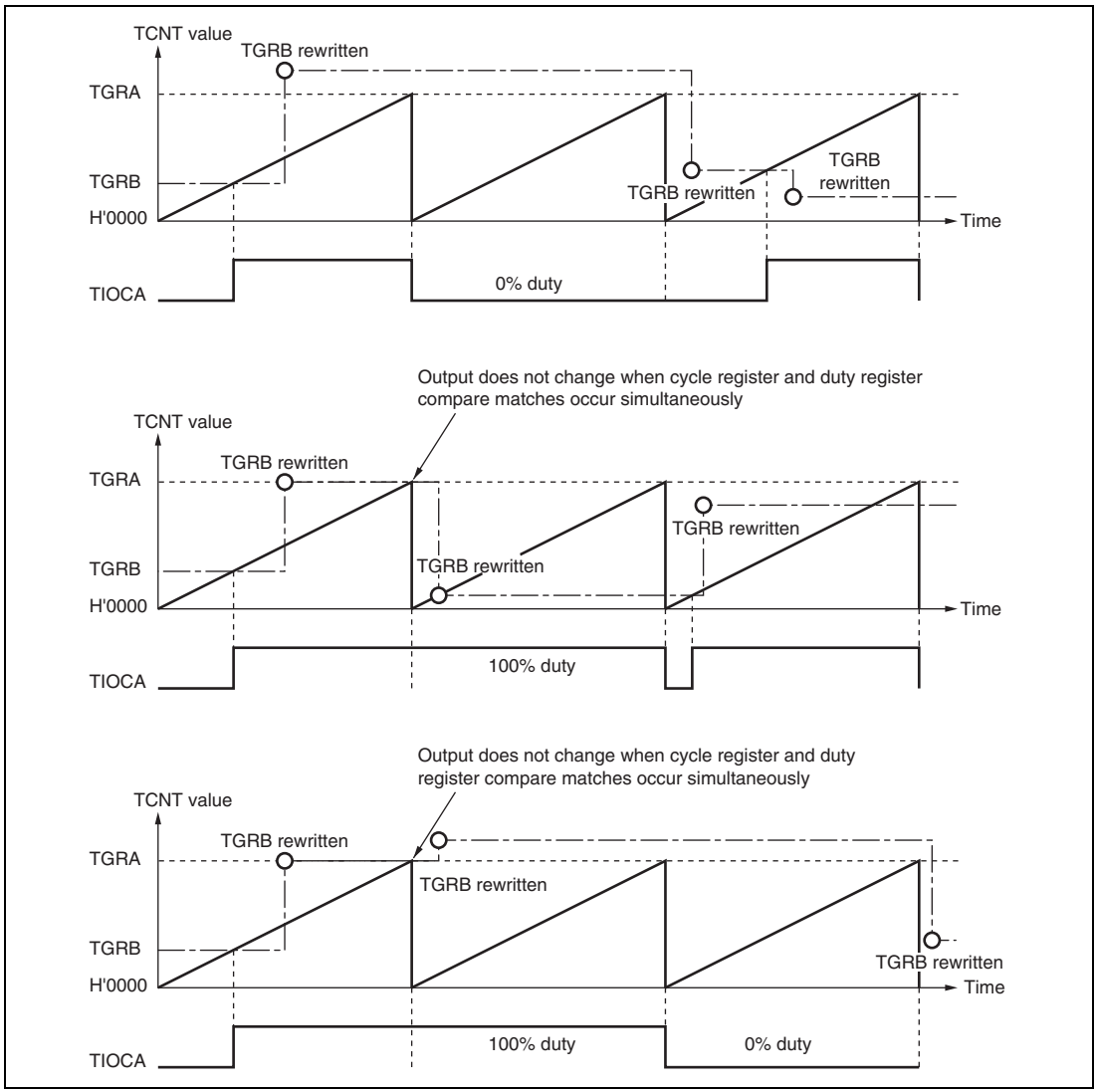



**Figure 10.28 Example of PWM Mode Operation (3)**

### 10.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1 and 2.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC0 to TPSC2 and bits CKEG0 and CKEG1 in TCR. However, the functions of bits CCLR0 and CCLR1 in TCR, and of TIOR, TIER, and TGR, are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflow occurs when TCNT is counting down, the TCFU flag is set.

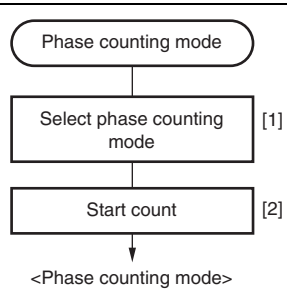
The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 10.47 shows the correspondence between external clock pins and channels.

**Table 10.47 Phase Counting Mode Clock Input Pins**

| | External Clock Pins | |
|---|---|---|
| **Channels** | **A-Phase** | **B-Phase** |
| When channel 1 is set to phase counting mode | TCLKA | TCLKB |
| When channel 2 is set to phase counting mode | TCLKC | TCLKD |

**Example of Phase Counting Mode Setting Procedure:** Figure 10.29 shows an example of the phase counting mode setting procedure.




[1] Select phase counting mode with bits MD3 to MD0 in TMDR.

[2] Set the CST bit in TSTR to 1 to start the count operation.

**Figure 10.29 Example of Phase Counting Mode Setting Procedure**

**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.

1. Phase counting mode 1

   Figure 10.30 shows an example of phase counting mode 1 operation, and table 10.48 summarizes the TCNT up/down-count conditions.

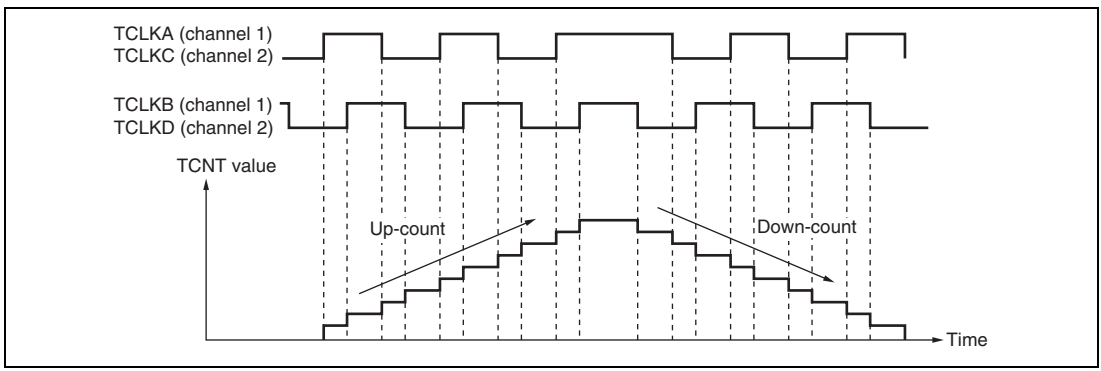



**Figure 10.30 Example of Phase Counting Mode 1 Operation**

**Table 10.48 Up/Down-Count Conditions in Phase Counting Mode 1**

| TCLKA (Channel 1)<br>TCLKC (Channel 2) | TCLKB (Channel 1)<br>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

2. Phase counting mode 2

   Figure 10.31 shows an example of phase counting mode 2 operation, and table 10.49 summarizes the TCNT up/down-count conditions.

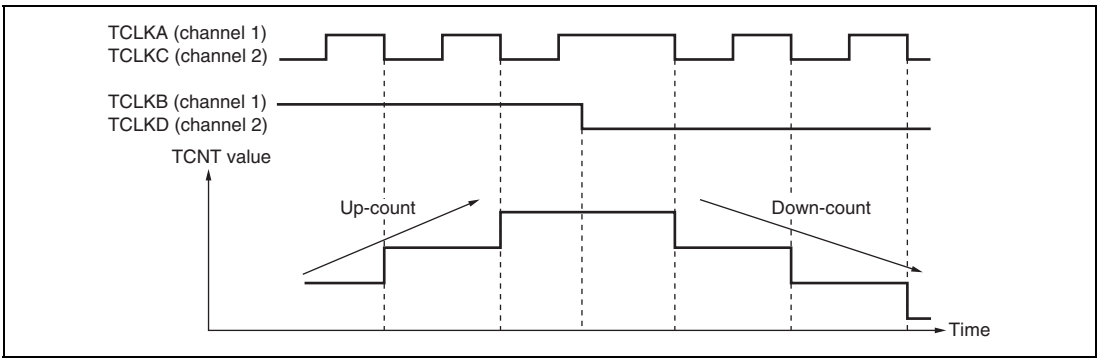



**Figure 10.31 Example of Phase Counting Mode 2 Operation**

**Table 10.49 Up/Down-Count Conditions in Phase Counting Mode 2**

| TCLKA (Channel 1)<br>TCLKC (Channel 2) | TCLKB (Channel 1)<br>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

3. Phase counting mode 3

   Figure 10.32 shows an example of phase counting mode 3 operation, and table 10.50 summarizes the TCNT up/down-count conditions.

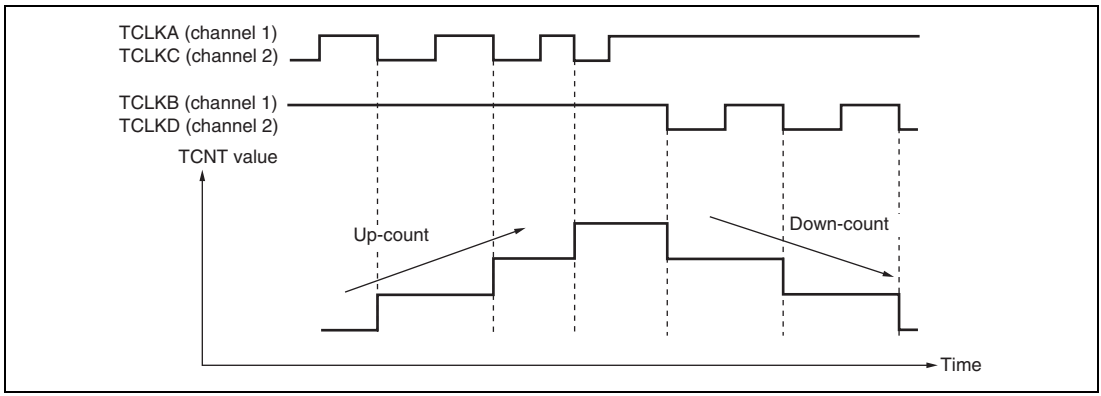



**Figure 10.32 Example of Phase Counting Mode 3 Operation**

**Table 10.50 Up/Down-Count Conditions in Phase Counting Mode 3**

| TCLKA (Channel 1)<br>TCLKC (Channel 2) | TCLKB (Channel 1)<br>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

4. Phase counting mode 4

   Figure 10.33 shows an example of phase counting mode 4 operation, and table 10.51 summarizes the TCNT up/down-count conditions.

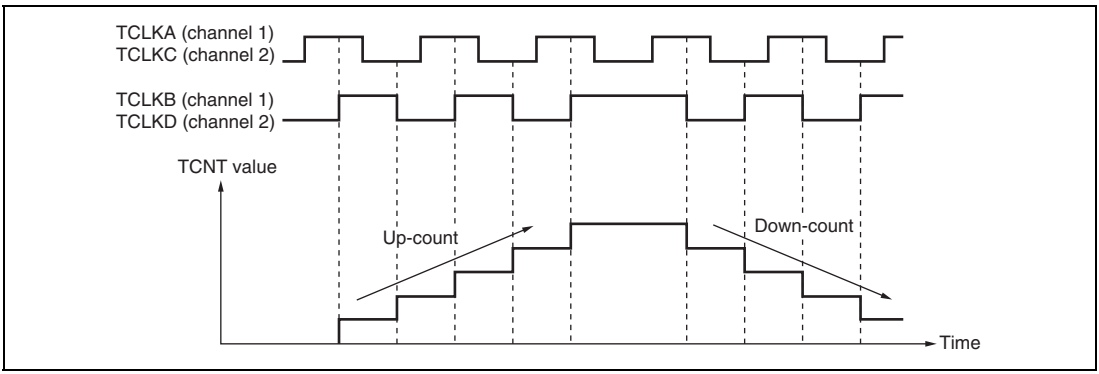



**Figure 10.33 Example of Phase Counting Mode 4 Operation**

**Table 10.51 Up/Down-Count Conditions in Phase Counting Mode 4**

| TCLKA (Channel 1)<br>TCLKC (Channel 2) | TCLKB (Channel 1)<br>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

**Phase Counting Mode Application Example:** Figure 10.34 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGRC_0 compare match; TGRA_0 and TGRC_0 are used for the compare match function and are set with the speed control period and position control period. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source and store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

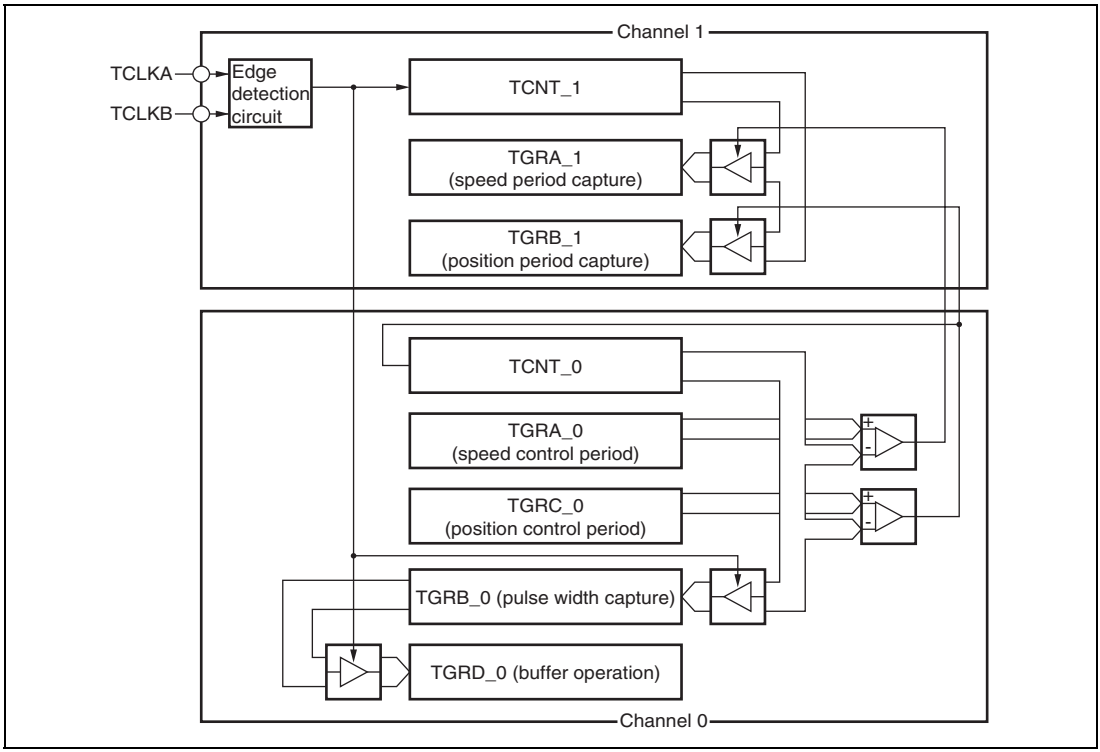



**Figure 10.34 Phase Counting Mode Application Example**

### 10.4.7 Reset-Synchronized PWM Mode

In 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 TCNT3 functions as an upcounter.

Table 10.52 shows the PWM output pins used. Table 10.53 shows the settings of the registers.

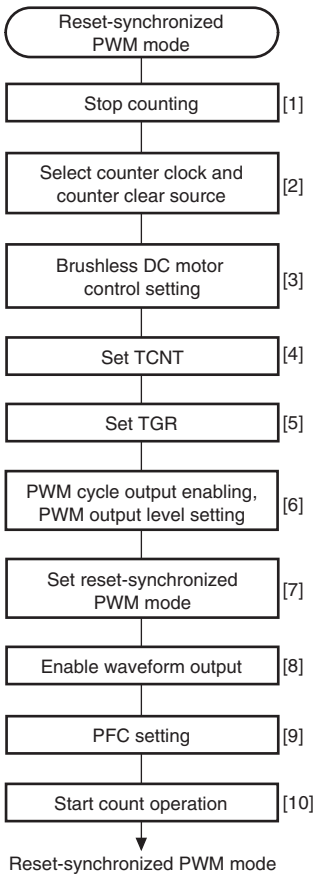
**Table 10.52 Output Pins for Reset-Synchronized PWM Mode**

| Channel | Output Pin | Description |
|---|---|---|
| 3 | TIOC3B | PWM output pin 1 |
| | TIOC3D | PWM output pin 1' (negative-phase waveform of PWM output 1) |
| 4 | TIOC4A | PWM output pin 2 |
| | TIOC4C | PWM output pin 2' (negative-phase waveform of PWM output 2) |
| | TIOC4B | PWM output pin 3 |
| | TIOC4D | PWM output pin 3' (negative-phase waveform of PWM output 3) |

**Table 10.53 Register Settings for Reset-Synchronized PWM Mode**

| Register | Description of Setting |
|---|---|
| TCNT_3 | Initial setting of H'0000 |
| TCNT_4 | Initial setting of H'0000 |
| TGRA_3 | Set count cycle for TCNT_3 |
| TGRB_3 | Sets the turning point for PWM waveform output by the TIOC3B and TIOC3D pins |
| TGRA_4 | Sets the turning point for PWM waveform output by the TIOC4A and TIOC4C pins |
| TGRB_4 | Sets the turning point for PWM waveform output by the TIOC4B and TIOC4D pins |

**Procedure for Selecting the Reset-Synchronized PWM Mode:** Figure 10.35 shows an example of procedure for selecting the reset synchronized PWM mode.




[1] Clear the CST3 and CST4 bits in the TSTR to 0 to halt the counting of TCNT. The reset-synchronized PWM mode must be set up while TCNT_3 and TCNT_4 are halted.

[2] Set bits TPSC2 to TPSC0 and CKEG1 and CKEG0 in the TCR_3 to select the counter clock and clock edge for channel 3. Set bits CCLR2 to CCLR0 in the TCR_3 to select TGRA compare-match as a counter clear source.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Reset TCNT_3 and TCNT_4 to H'0000.

[5] TGRA_3 is the period register. Set the waveform period value in TGRA_3. Set the transition timing of the PWM output waveforms in TGRB_3, TGRA_4, and TGRB_4. Set times within the compare-match range of TCNT_3.
$X \leq$ TGRA_3 (X: set value).

[6] Select enabling/disabling of toggle output synchronized with the PMW cycle using bit PSYE in the timer output control register (TOCR1), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR2, see figure 10.3.

[7] Set bits MD3 to MD0 in TMDR_3 to B'1000 to select the reset-synchronized PWM mode. Do not set to TMDR_4.

[8] Set the enabling/disabling of the PWM waveform output pin in TOER.

[9] Set the port control register and the port I/O register.

[10] Set the CST3 bit in the TSTR to 1 to start the count operation.

Note: The output waveform starts to toggle operation at the point of TCNT_3 = TGRA_3 = X by setting X = TGRA, i.e., cycle = duty.

**Figure 10.35 Procedure for Selecting Reset-Synchronized PWM Mode**

**Reset-Synchronized PWM Mode Operation:** Figure 10.36 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.

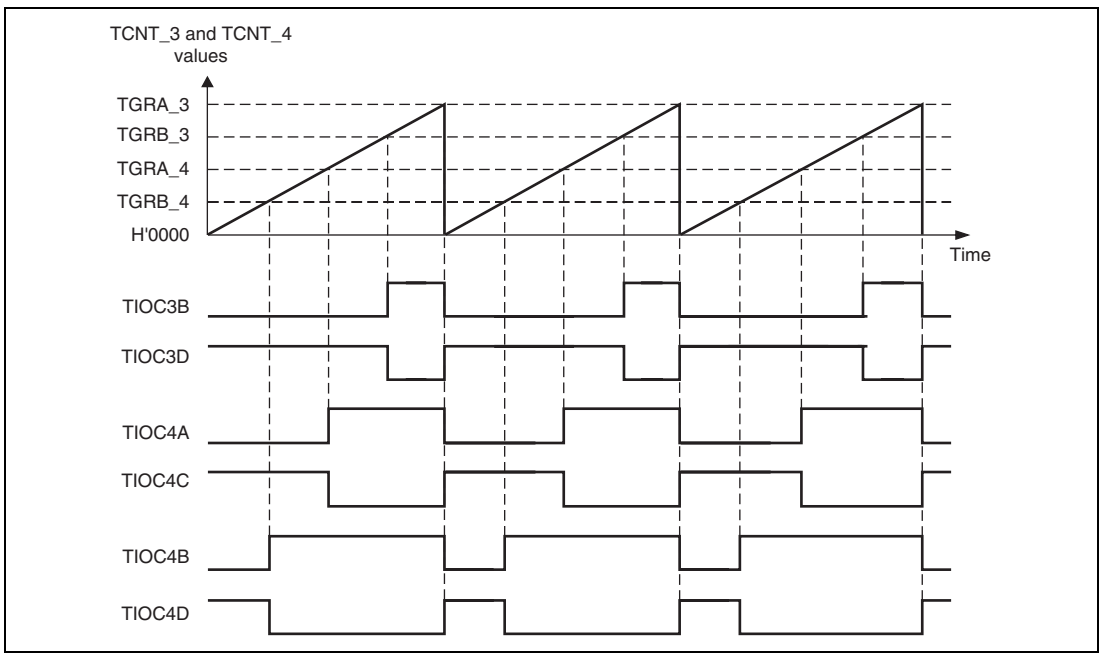



**Figure 10.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)**

### 10.4.8 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 10.54 shows the PWM output pins used. Table 10.55 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

**Table 10.54 Output Pins for Complementary PWM Mode**

| Channel | Output Pin | Description |
|---|---|---|
| 3 | TIOC3A | Toggle output synchronized with PWM period (or I/O port) |
| | TIOC3B | PWM output pin 1 |
| | TIOC3C | I/O port* |
| | TIOC3D | PWM output pin 1'<br>(non-overlapping negative-phase waveform of PWM output 1;<br>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'<br>(non-overlapping negative-phase waveform of PWM output 2;<br>PWM output without non-overlapping interval is also available) |
| | TIOC4D | PWM output pin 3'<br>(non-overlapping negative-phase waveform of PWM output 3;<br>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 10.55 Register Settings for Complementary PWM Mode**

| Channel | Counter/Register | Description | Read/Write from CPU |
|---|---|---|---|
| 3 | TCNT_3 | Start of up-count from value set in dead time register | Maskable by TRWER setting* |
| | TGRA_3 | Set TCNT_3 upper limit value (1/2 carrier cycle + dead time) | Maskable by TRWER setting* |
| | TGRB_3 | PWM output 1 compare register | Maskable by TRWER setting* |
| | TGRC_3 | TGRA_3 buffer register | Always readable/writable |
| | TGRD_3 | PWM output 1/TGRB_3 buffer register | Always readable/writable |
| 4 | TCNT_4 | Up-count start, initialized to H'0000 | Maskable by TRWER setting* |
| | TGRA_4 | PWM output 2 compare register | Maskable by TRWER setting* |
| | TGRB_4 | PWM output 3 compare register | Maskable by TRWER setting* |
| | TGRC_4 | PWM output 2/TGRA_4 buffer register | Always readable/writable |
| | TGRD_4 | PWM output 3/TGRB_4 buffer register | Always readable/writable |
| Timer dead time data register (TDDR) | | Set TCNT_4 and TCNT_3 offset value (dead time value) | Maskable by TRWER setting* |
| Timer cycle data register (TCDR) | | Set TCNT_4 upper limit value (1/2 carrier cycle) | Maskable by TRWER setting* |
| Timer cycle buffer register (TCBR) | | TCDR buffer register | Always readable/writable |
| Subcounter (TCNTS) | | Subcounter for dead time generation | Read-only |
| Temporary register 1 (TEMP1) | | PWM output 1/TGRB_3 temporary register | Not readable/writable |
| Temporary register 2 (TEMP2) | | PWM output 2/TGRA_4 temporary register | Not readable/writable |
| Temporary register 3 (TEMP3) | | PWM output 3/TGRB_4 temporary register | Not readable/writable |

Note: * Access can be enabled or disabled according to the setting of bit 0 (RWE) in TRWER (timer read/write enable register).

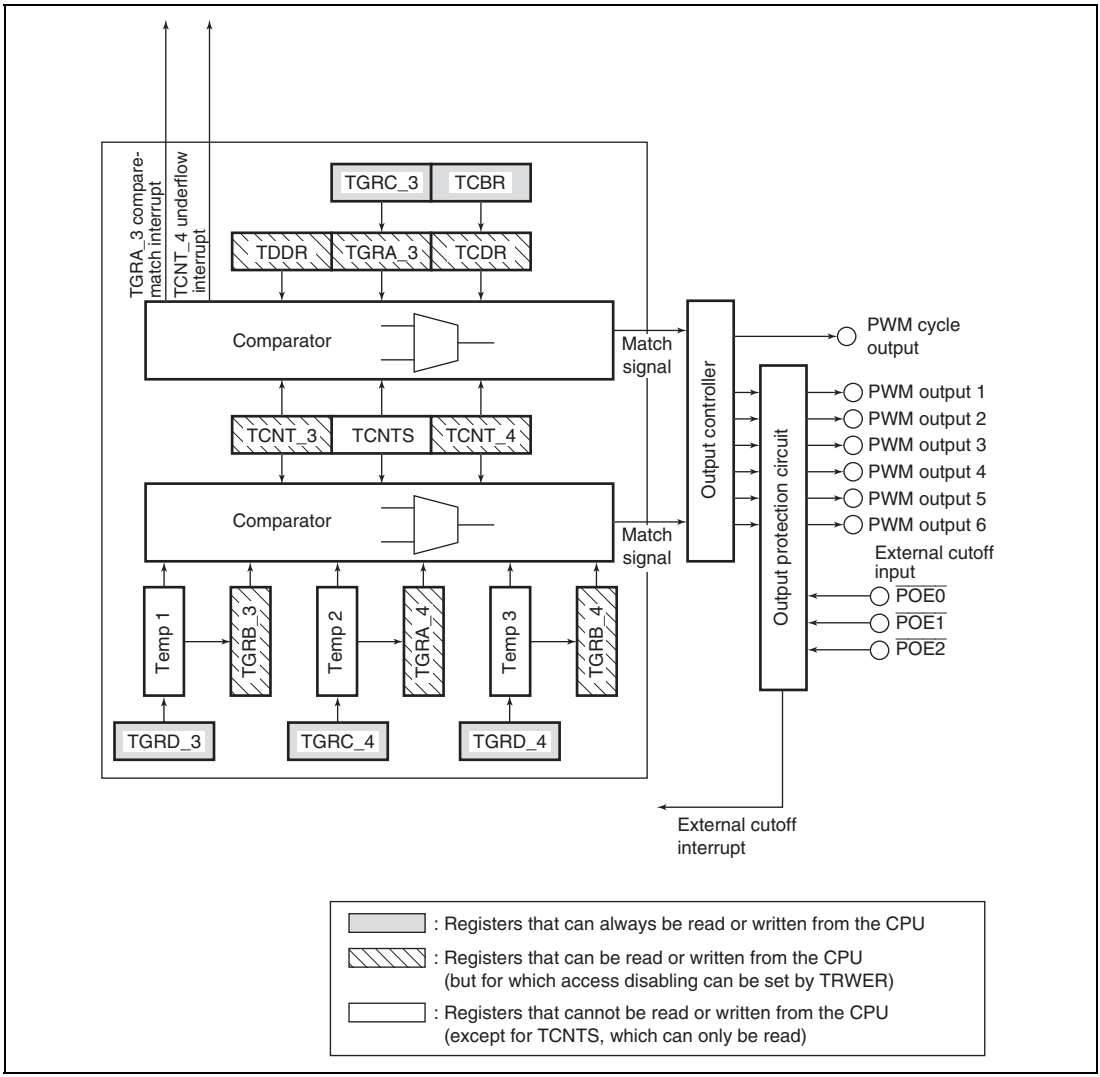
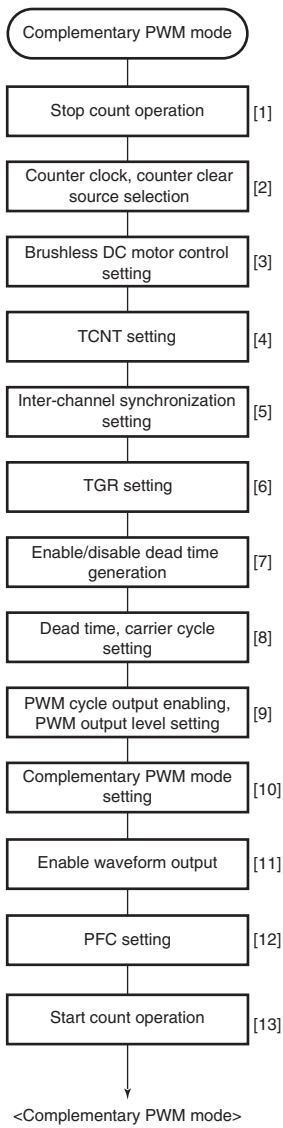



**Figure 10.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode**

**Example of Complementary PWM Mode Setting Procedure:** An example of the complementary PWM mode setting procedure is shown in figure 10.38.




[1] Clear bits CST3 and CST4 in the timer start register (TSTR) to 0, and halt timer counter (TCNT) operation. Perform complementary PWM mode setting when TCNT_3 and TCNT_4 are stopped.

[2] Set the same counter clock and clock edge for channels 3 and 4 with bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in the timer control register (TCR). Use bits CCLR2 to CCLR0 to set synchronous clearing only when restarting by a synchronous clear from another channel during complementary PWM mode operation.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set the dead time in TCNT_3. Set TCNT_4 to H'0000.

[5] Set only when restarting by a synchronous clear from another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with channels 3 and 4 using the timer synchro register (TSYR).

[6] Set the output PWM duty in the duty registers (TGRB_3, TGRA_4, TGRB_4) and buffer registers (TGRD_3, TGRC_4, TGRD_4). Set the same initial value in each corresponding TGR.

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the timer dead time enable register (TDER) so that no dead time is generated.

[8] Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the carrier cycle data register (TCDR) and carrier cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in TGRA_3 and TGRC_3. When no dead time generation is selected, set 1 in TDDR and 1/2 the carrier cycle + 1 in TGRA_3 and TGRC_3.

[9] Select enabling/disabling of toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 10.3.

[10] Select complementary PWM mode in timer mode register 3 (TMDR_3). Do not set in TMDR_4.

[11] Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).

[12] Set the port control register and the port I/O register.

[13] Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

**Figure 10.38 Example of Complementary PWM Mode Setting Procedure**

**Outline of Complementary PWM Mode Operation:**

In complementary PWM mode, 6-phase PWM output is possible. Figure 10.39 illustrates counter operation in complementary PWM mode, and figure 10.40 shows an example of complementary PWM mode operation.

1. Counter Operation

   In complementary PWM mode, three counters—TCNT_3, TCNT_4, and TCNTS—perform up/down-count operations.

   TCNT_3 is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

   When the CST bit is set to 1, TCNT_3 counts up to the value set in TGRA_3, then switches to down-counting when it matches TGRA_3. When the TCNT3 value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

   TCNT_4 is initialized to H'0000.

   When the CST bit is set to 1, TCNT4 counts up in synchronization with TCNT_3, and switches to down-counting when it matches TCDR. On reaching H'0000, TCNT4 switches to up-counting, and the operation is repeated in this way.

   TCNTS is a read-only counter. It need not be initialized.

   When TCNT_3 matches TCDR during TCNT_3 and TCNT_4 up/down-counting, down-counting is started, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches TGRA_3, it is cleared to H'0000.

   When TCNT_4 matches TDDR during TCNT_3 and TCNT_4 down-counting, up-counting is started, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches H'0000, it is set with the value in TGRA_3.

   TCNTS is compared with the compare register and temporary register in which the PWM duty is set during the count operation only.

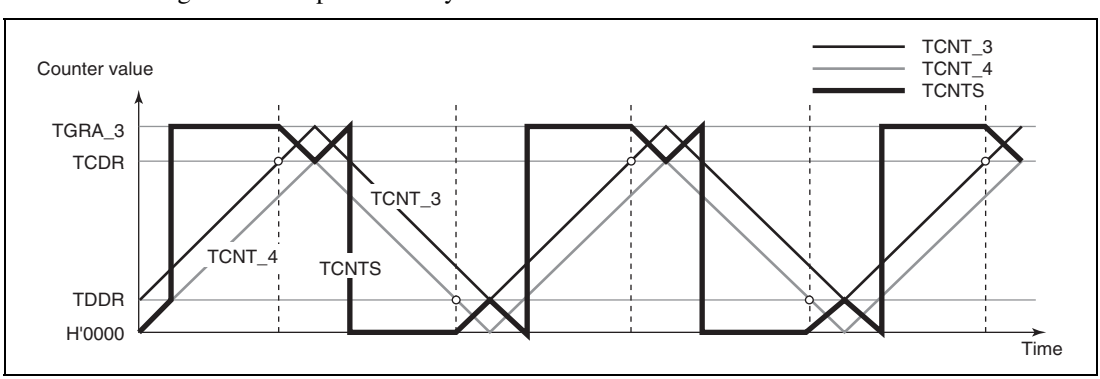



**Figure 10.39 Complementary PWM Mode Counter Operation**

2. Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 10.40 shows an example of complementary PWM mode operation.

The registers which are constantly compared with the counters to perform PWM output are TGRB_3, TGRA_4, and TGRB_4. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register (TOCR) is output.

The buffer registers for these compare registers are TGRD_3, TGRC_4, and TGRD_4.

Between a buffer register and compare register there is a temporary register. The temporary registers cannot be accessed by the CPU.

Data in a compare register is changed by writing the new data to the corresponding buffer register. The buffer registers can be read or written at any time.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches TGRA_3 when counting up, or H'0000 when counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD3 to MD0 in the timer mode register (TMDR). Figure 10.40 shows an example in which the mode is selected in which the change is made in the trough.

In the Tb interval (Tb1 in figure 10.40) in which data transfer to the temporary register is not performed, the temporary register has the same function as the compare register, and is compared with the counter. In this interval, therefore, there are two compare match registers for one-phase output, with the compare register containing the pre-change data, and the temporary register containing the new data. In this interval, the three counters—TCNT_3, TCNT_4, and TCNTS—and two registers—compare register and temporary register—are compared, and PWM output controlled accordingly.

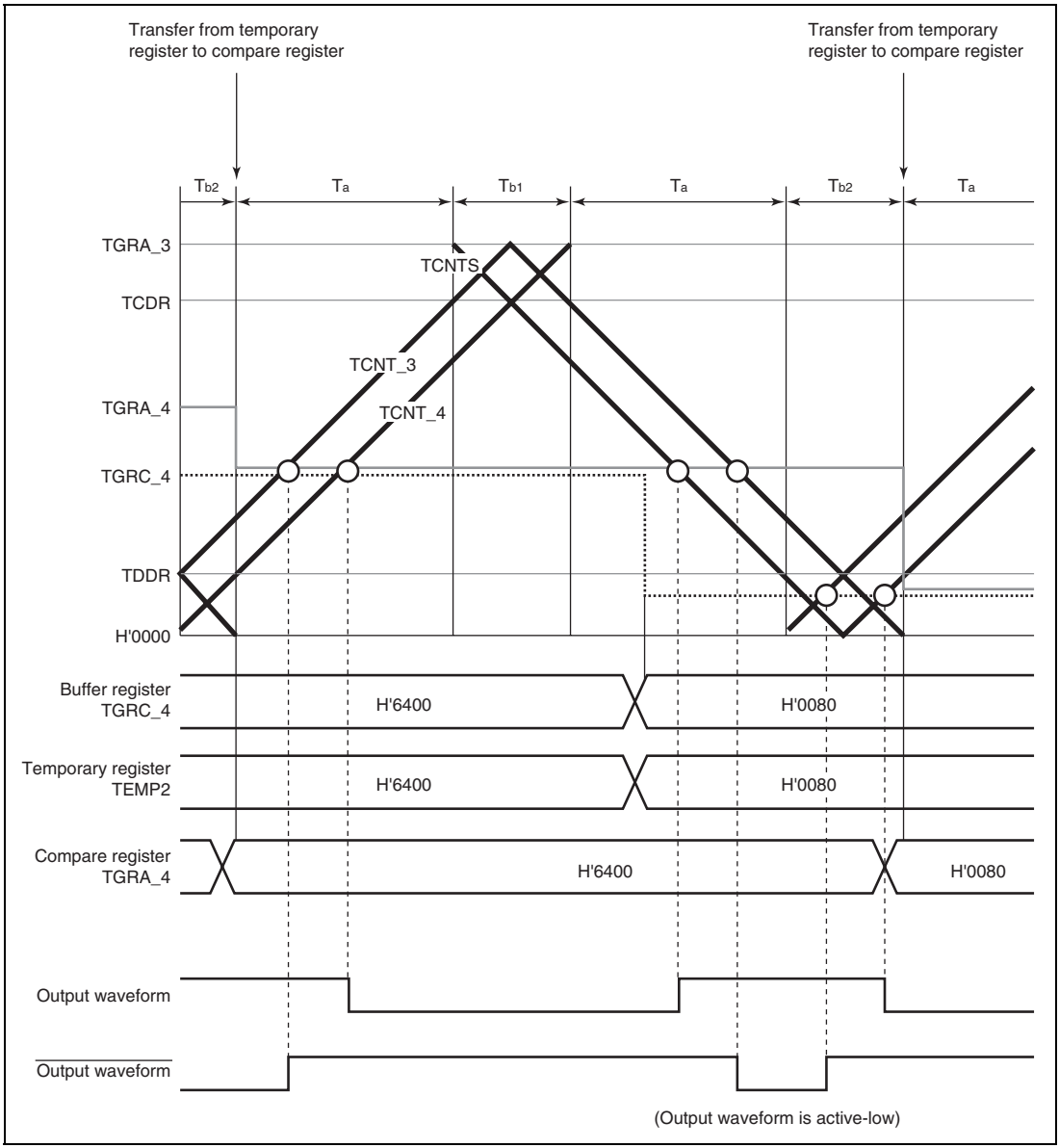



**Figure 10.40 Example of Complementary PWM Mode Operation**

3. Initialization

   In complementary PWM mode, there are six registers that must be initialized. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).

   Before setting complementary PWM mode with bits MD3 to MD0 in the timer mode register (TMDR), the following initial register values must be set.

   TGRC_3 operates as the buffer register for TGRA_3, and should be set with 1/2 the PWM carrier cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time Td in the timer dead time data register (TDDR).

   When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, TGRC_3 and TGRA_3 should be set to 1/2 the PWM carrier cycle + 1, and TDDR should be set to 1.

   Set the respective initial PWM duty values in buffer registers TGRD_3, TGRC_4, and TGRD_4.

   The values set in the five buffer registers excluding TDDR are transferred simultaneously to the corresponding compare registers when complementary PWM mode is set.

   Set TCNT_4 to H'0000 before setting complementary PWM mode.

**Table 10.56 Registers and Counters Requiring Initialization**

| Register/Counter | Set Value |
|---|---|
| TGRC_3 | 1/2 PWM carrier cycle + dead time Td<br>(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.

4. 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.

5. 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.

6. Dead Time Suppressing

   Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDER) to 0. TDER can be cleared to 0 only when 0 is written to it after reading TDER = 1.

   TGRA_3 and TGRC_3 should be set to 1/2 PWM carrier cycle + 1 and the timer dead time data register (TDDR) should be set to 1.

   By the above settings, PWM waveforms without dead time can be obtained. Figure 10.41 shows an example of operation without dead time.

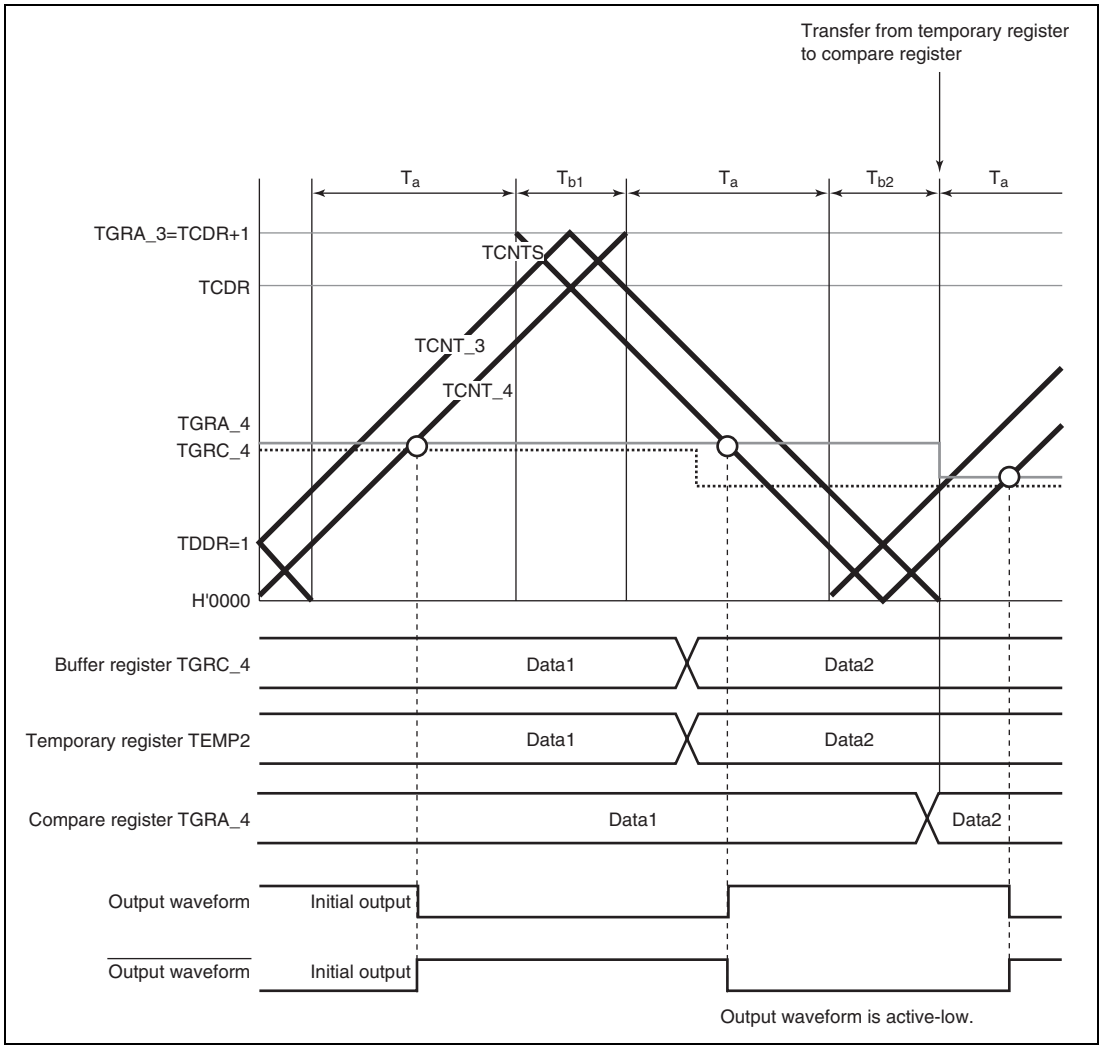



**Figure 10.41 Example of Operation without Dead Time**

7. PWM Cycle Setting

In complementary PWM mode, the PWM pulse cycle is set in two registers—TGRA_3, in which the TCNT_3 upper limit value is set, and TCDR, in which the TCNT_4 upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: TGRA_3 set value = TCDR set value + TDDR set value
Without dead time: TGRA_3 set value = TCDR set value + 1

The TGRA_3 and TCDR settings are made by setting the values in buffer registers TGRC_3 and TCBR. The values set in TGRC_3 and TCBR are transferred simultaneously to TGRA_3 and TCDR in accordance with the transfer timing selected with bits MD3 to MD0 in the timer mode register (TMDR).

The updated PWM cycle is reflected from the next cycle when the data update is performed at the crest, and from the current cycle when performed in the trough. Figure 10.42 illustrates the operation when the PWM cycle is updated at the crest.

See the following section, Register Data Updating, for the method of updating the data in each buffer register.

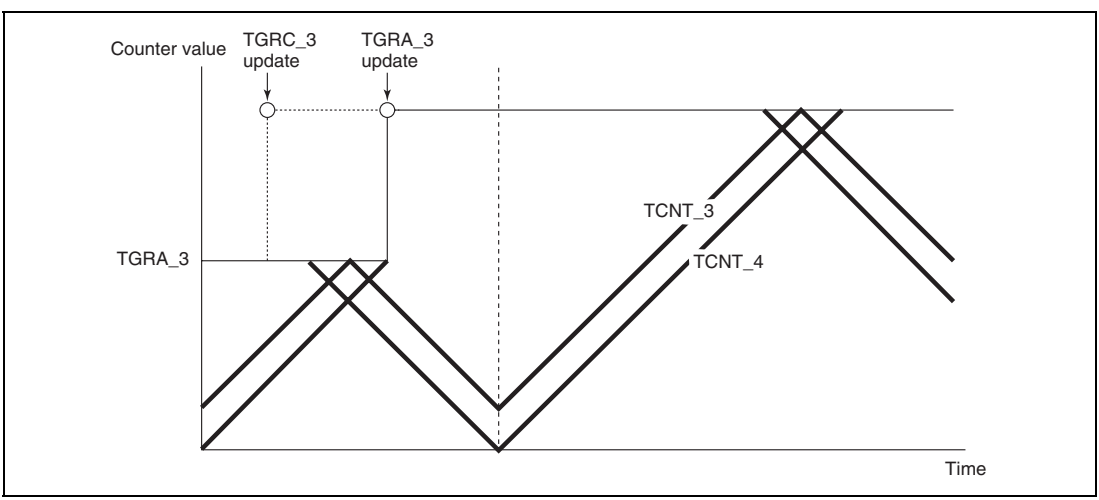



**Figure 10.42 Example of PWM Cycle Updating**

8. Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five PWM duty and carrier cycle registers that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. When subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value is also rewritten. Transfer is not performed from buffer registers to temporary registers when TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD3 to MD0 in the timer mode register (TMDR). Figure 10.43 shows an example of data updating in complementary PWM mode. This example shows the mode in which data updating is performed at both the counter crest and trough.

When rewriting buffer register data, a write to TGRD_4 must be performed at the end of the update. Data transfer from the buffer registers to the temporary registers is performed simultaneously for all five registers after the write to TGRD_4.

A write to TGRD_4 must be performed after writing data to the registers to be updated, even when not updating all five registers, or when updating the TGRD_4 data. In this case, the data written to TGRD_4 should be the same as the data prior to the write operation.

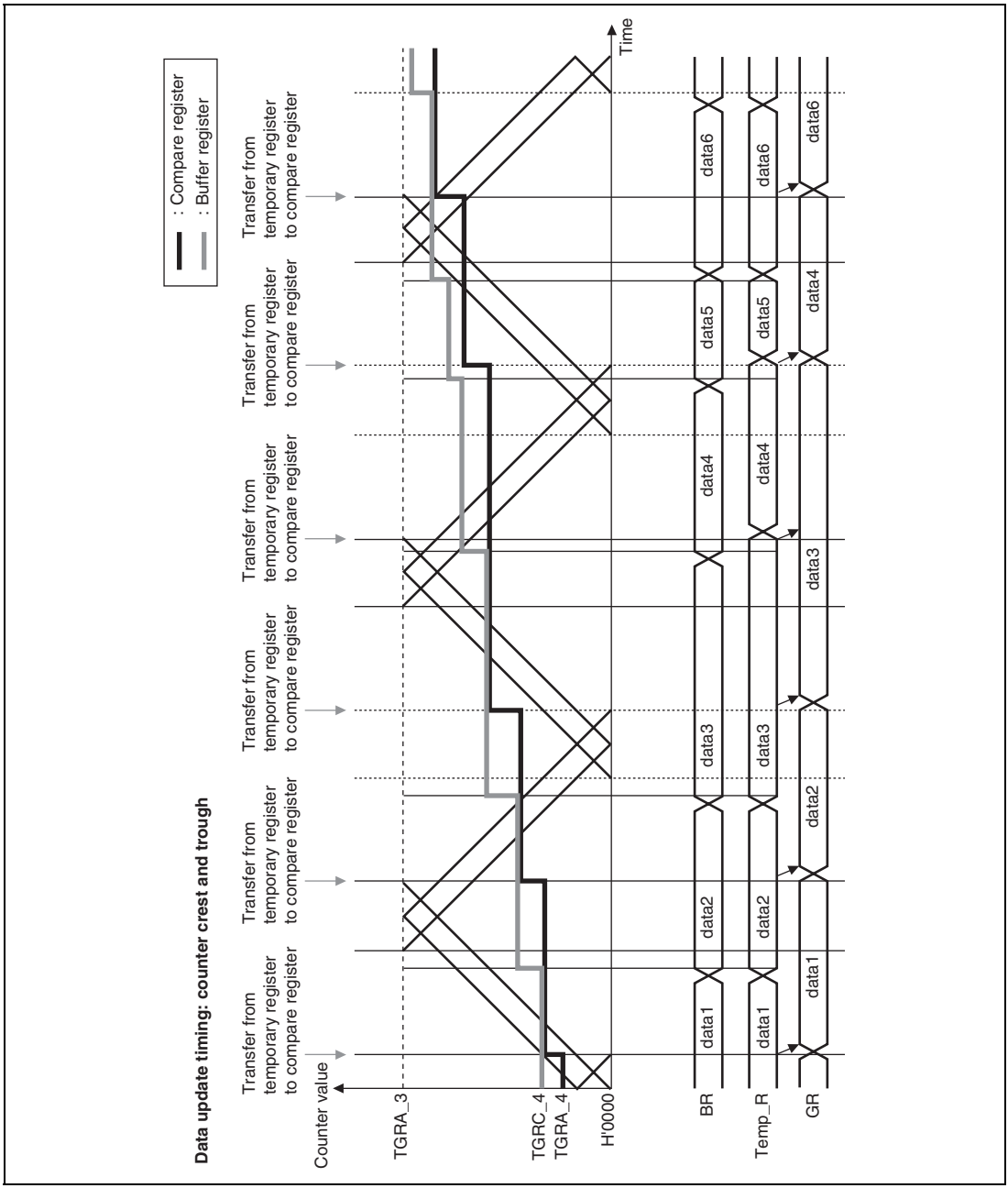



**Figure 10.43 Example of Data Update in Complementary PWM Mode**

9. Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2).

This initial output is the PWM pulse non-active level, and is output from when complementary PWM mode is set with the timer mode register (TMDR) until TCNT_4 exceeds the value set in the dead time register (TDDR). Figure 10.44 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in figure 10.45.

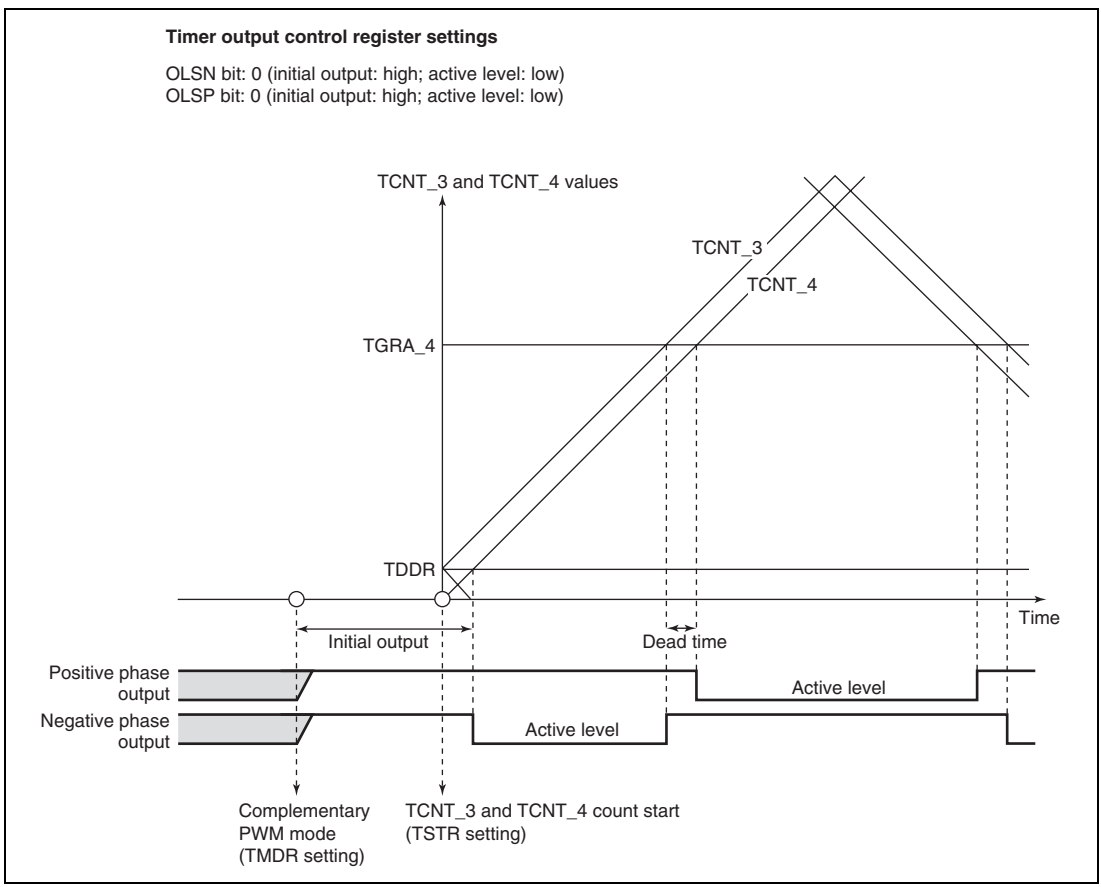



**Figure 10.44 Example of Initial Output in Complementary PWM Mode (1)**

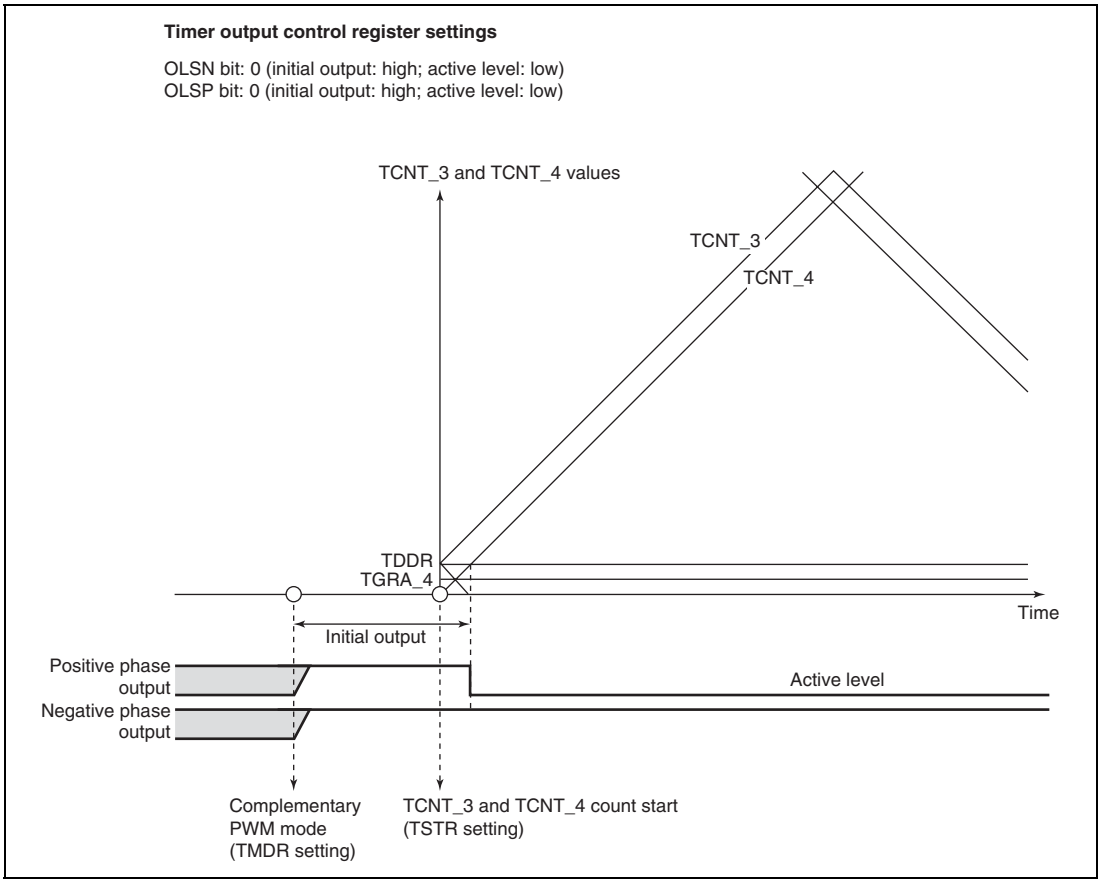



**Figure 10.45 Example of Initial Output in Complementary PWM Mode (2)**

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 10.46 to 10.48 show examples of waveform generation in complementary PWM mode.

The positive phase/negative phase off timing is generated by a compare-match with the solid-line counter, and the on timing by a compare-match with the dotted-line counter operating with a delay of the dead time behind the solid-line counter. In the T1 period, compare-match **a** that turns off the negative phase has the highest priority, and compare-matches occurring prior to **a** are ignored. In the T2 period, compare-match **c** that turns off the positive phase has the highest priority, and compare-matches occurring prior to **c** are ignored.

In normal cases, compare-matches occur in the order **a** → **b** → **c** → **d** (or **c** → **d** → **a'** → **b'**), as shown in figure 10.46.

If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which the negative phase is off is less than twice the dead time, the figure shows the positive phase is not being turned on. If compare-matches deviate from the **c** → **d** → **a'** → **b'** order, since the time for which the positive phase is off is less than twice the dead time, the figure shows the negative phase is not being turned on.

If compare-match **c** occurs first following compare-match **a**, as shown in figure 10.47, compare-match **b** is ignored, and the negative phase is turned off by compare-match **d**. This is because turning off of the positive phase has priority due to the occurrence of compare-match **c** (positive phase off timing) before compare-match **b** (positive phase on timing) (consequently, the waveform does not change since the positive phase goes from off to off).

Similarly, in the example in figure 10.48, compare-match **a'** with the new data in the temporary register occurs before compare-match **c**, but other compare-matches occurring up to **c**, which turns off the positive phase, are ignored. As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare-matches at turn-off timings take precedence, and turn-on timing compare-matches that occur before a turn-off timing compare-match are ignored.

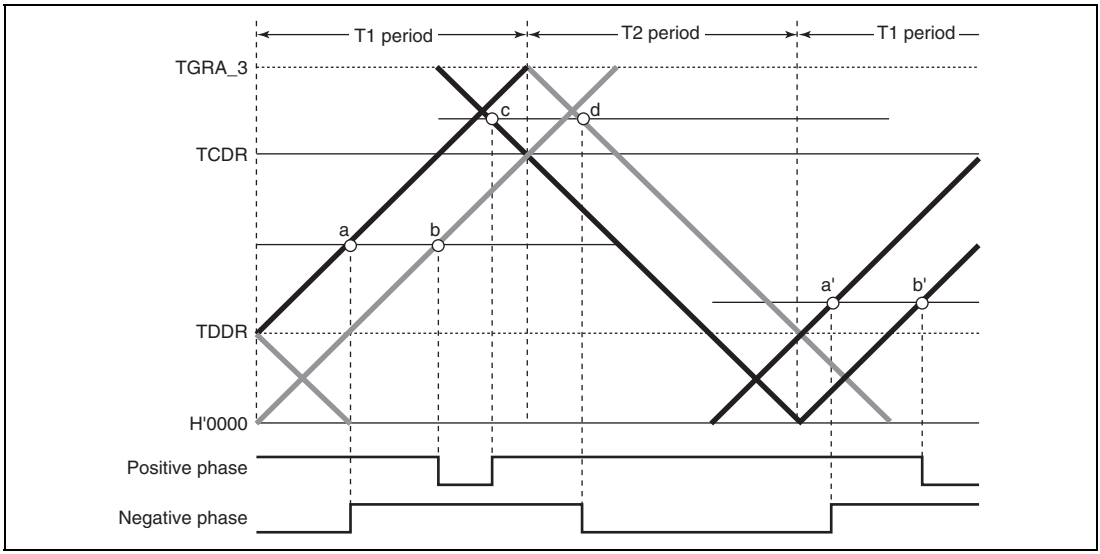



**Figure 10.46 Example of Complementary PWM Mode Waveform Output (1)**

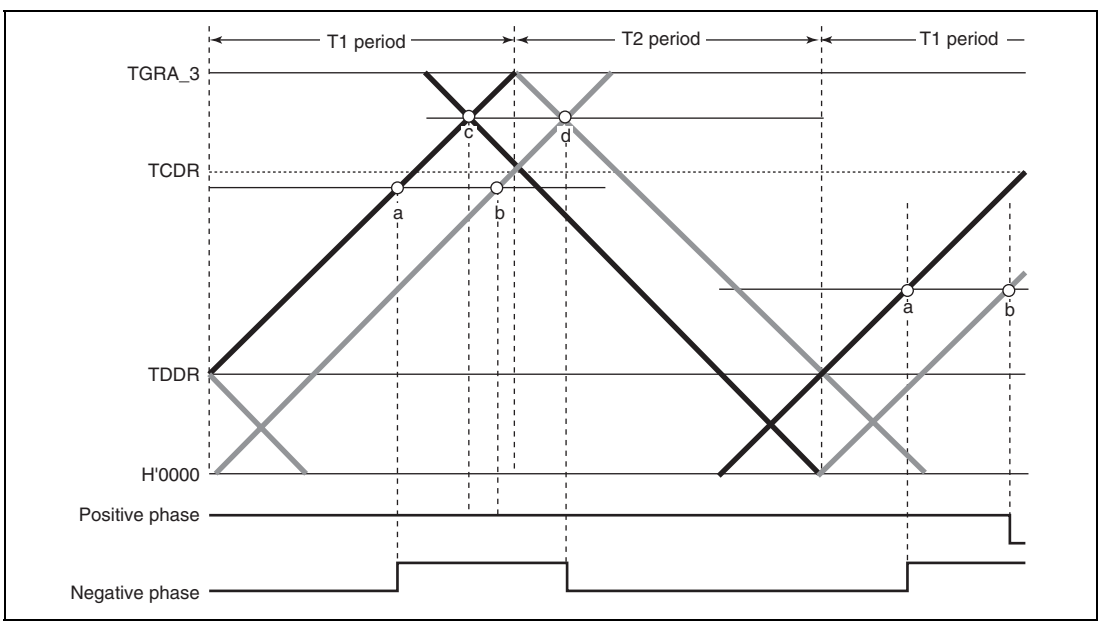



**Figure 10.47 Example of Complementary PWM Mode Waveform Output (2)**

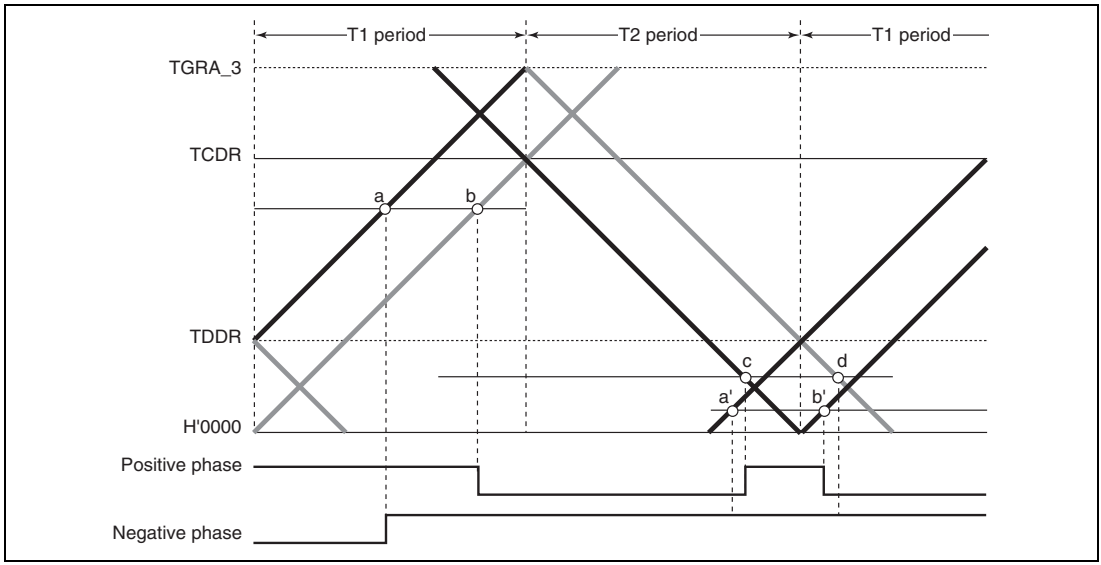



**Figure 10.48 Example of Complementary PWM Mode Waveform Output (3)**

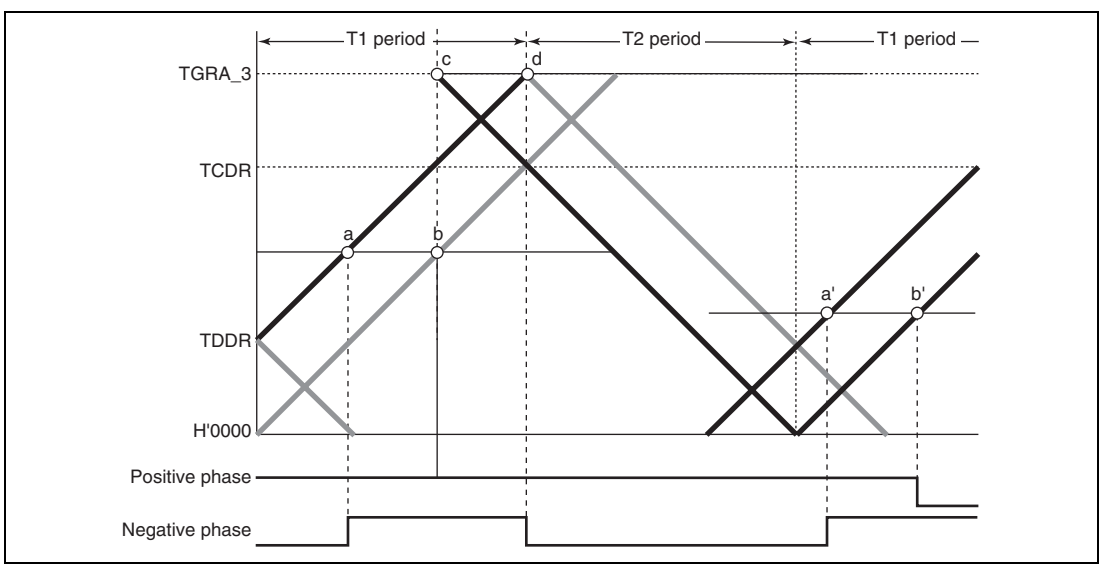



**Figure 10.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)**

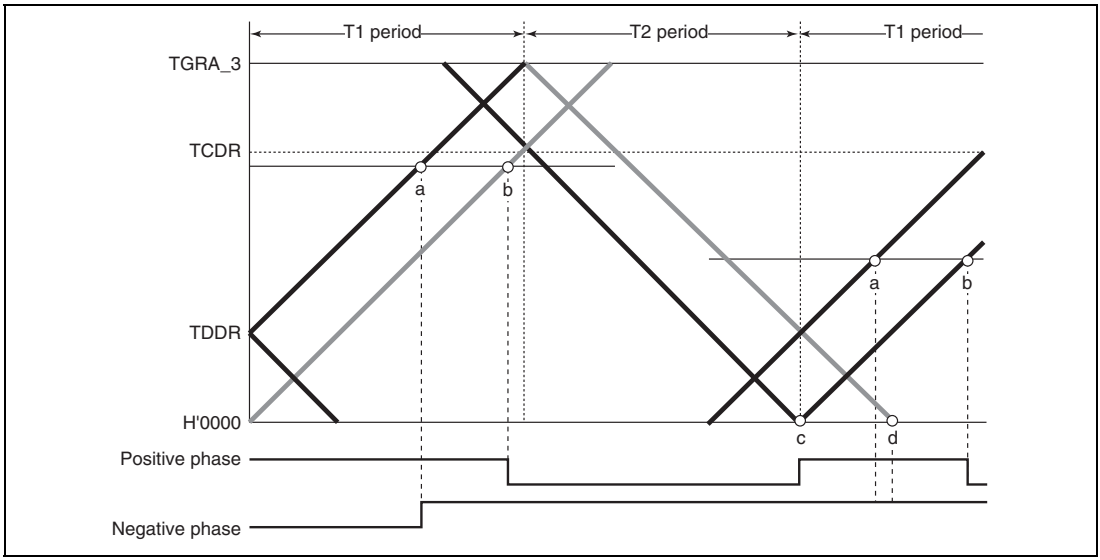



**Figure 10.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)**

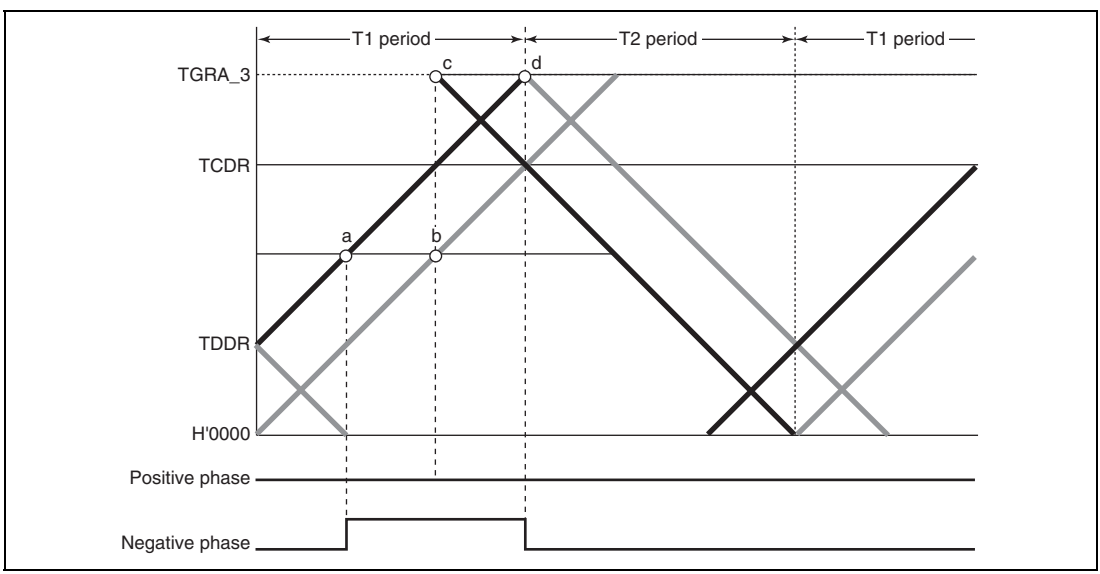



**Figure 10.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)**

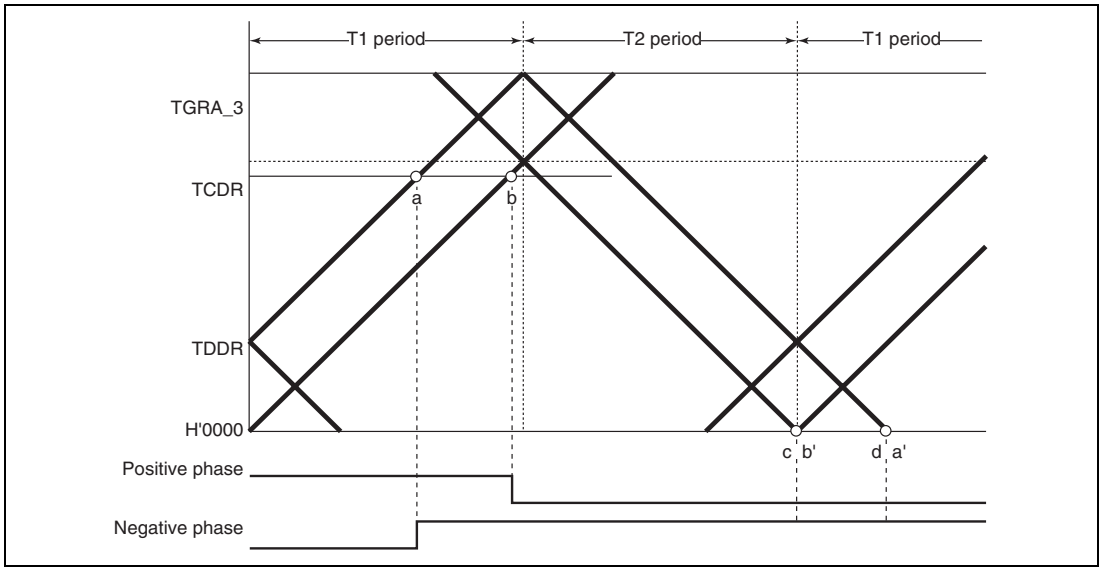



**Figure 10.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)**

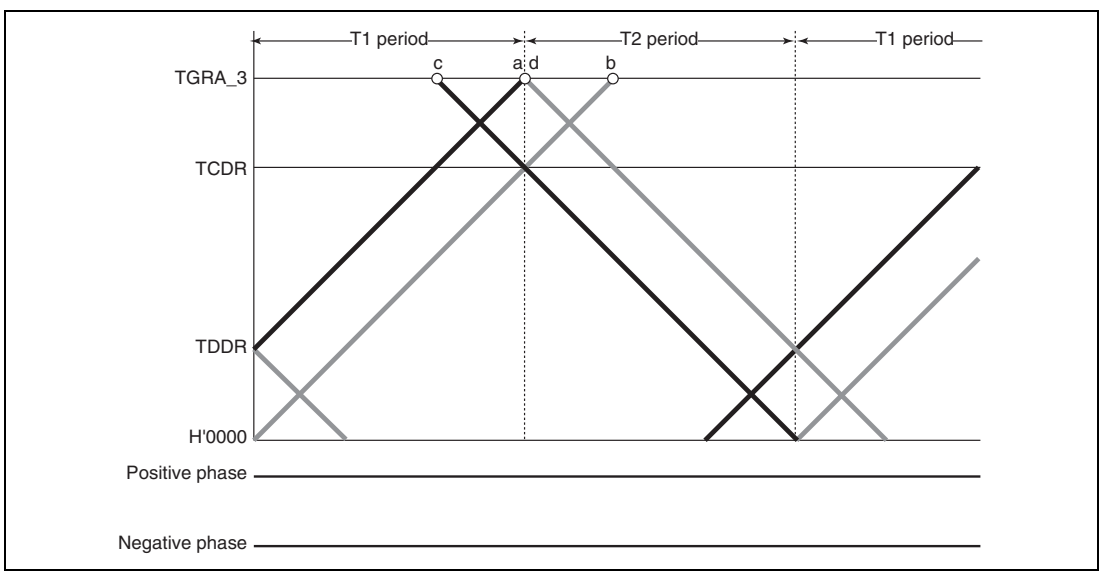



**Figure 10.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)**

11. Complementary PWM Mode 0% and 100% Duty Output

    In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 10.49 to 10.53 show output examples.

    100% duty output is performed when the data register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the data register value is set to the same value as TGRA_3. The waveform in this case has a positive phase with a 100% off-state.

    On and off compare-matches occur simultaneously, but if a turn-on compare-match and turn-off compare-match for the same phase occur simultaneously, both compare-matches are ignored and the waveform does not change.

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 10.54.

    This output is toggled by a compare-match between TCNT_3 and TGRA_3 and a compare-match between TCNT4 and H'0000.

    The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

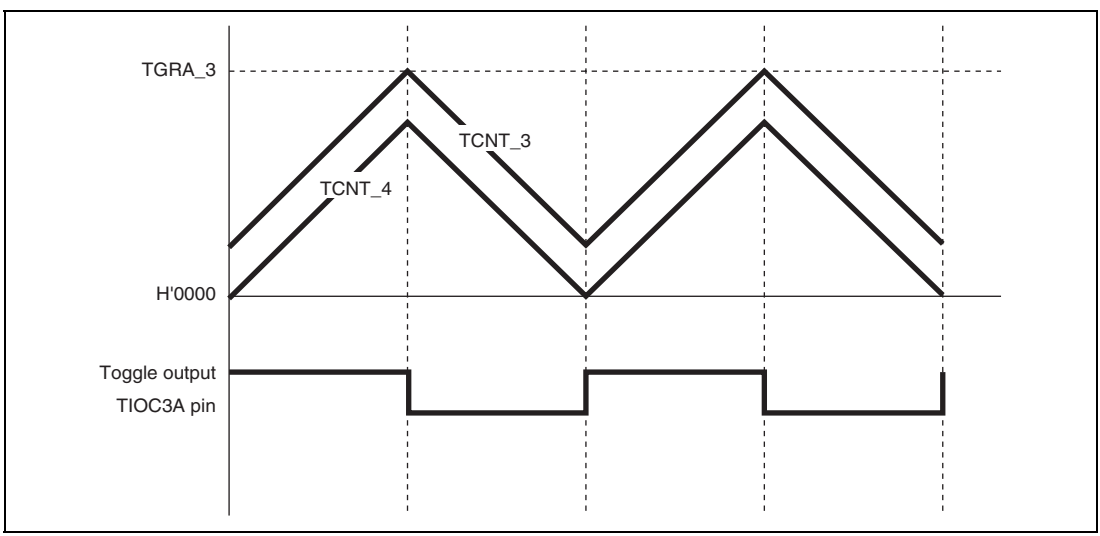



**Figure 10.54 Example of Toggle Output Waveform Synchronized with PWM Output**

13. Counter Clearing by Another Channel

In complementary PWM mode, by setting a mode for synchronization with another channel by means of the timer synchronous register (TSYR), and selecting synchronous clearing with bits CCLR2 to CCLR0 in the timer control register (TCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by another channel.

Figure 10.55 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

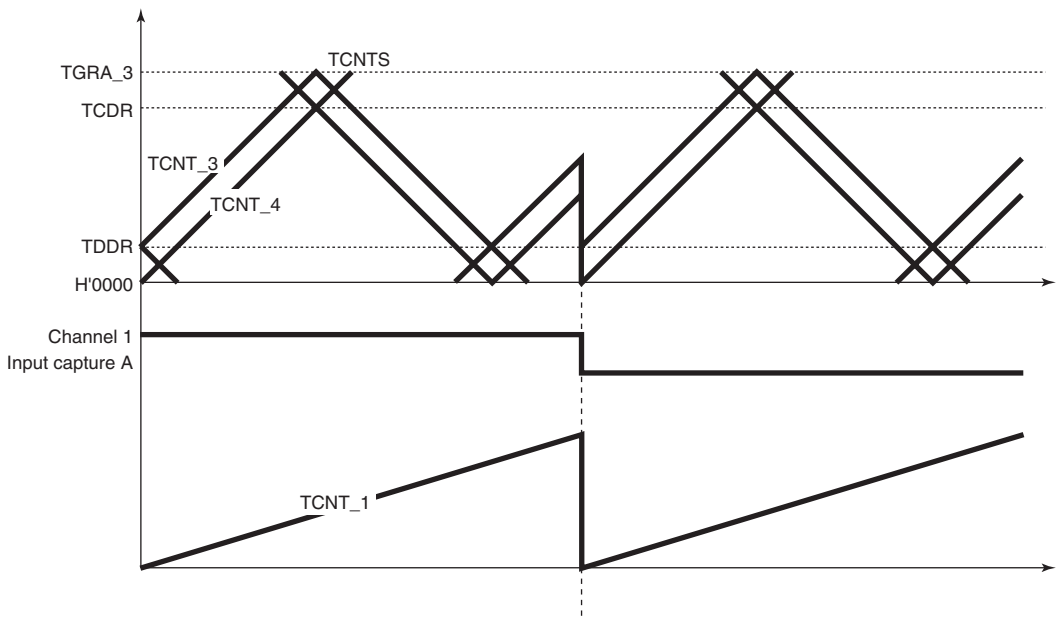



**Figure 10.55 Counter Clearing Synchronized with Another Channel**

14. Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCR to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in figure 10.56. When synchronous clearing occurs outside that interval, the initial value specified by the OLS bits in TOCR is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial value output period (indicated by (1) in figure 10.56) immediately after the counters start operation, initial value output is not suppressed.

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 10.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.

This function can be used in both the MTU2 and MTU2S. In the MTU2, synchronous clearing generated in channels 0 to 2 in the MTU2 can cause counter clearing in complementary PWM mode; in the MTU2S, compare match or input capture flag setting in channels 0 to 2 in the MTU2 can cause counter clearing.

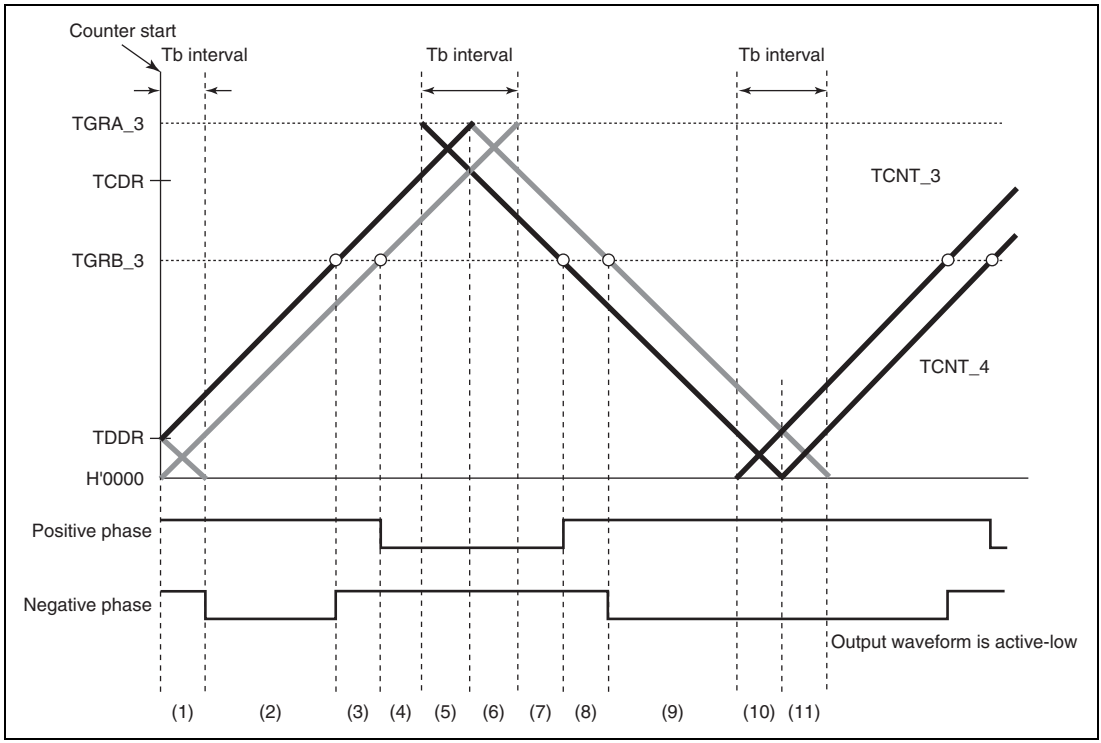



**Figure 10.56 Timing for Synchronous Counter Clearing**

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

An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in figure 10.57.

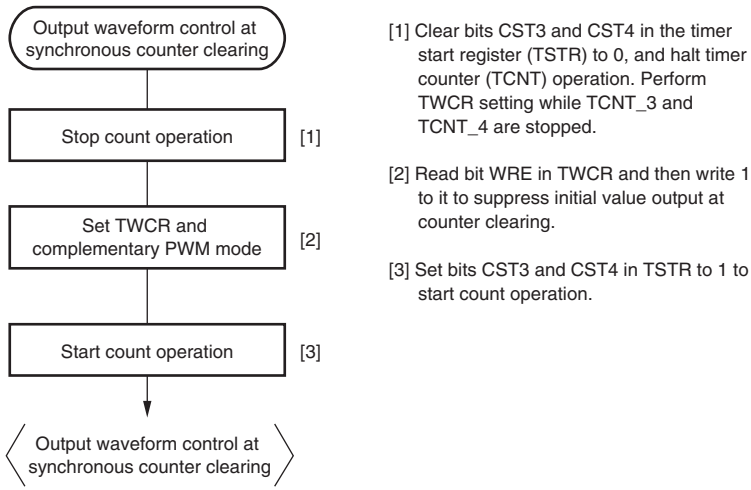



**Figure 10.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode**

— Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figures 10.58 to 10.61 show examples of output waveform control in which the MTU2 operates in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCR is set to 1. In the examples shown in figures 10.58 to 10.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 10.56, respectively.

In the MTU2S, these examples are equivalent to the cases when the MTU2S operates in complementary PWM mode and synchronous counter clearing is generated while the SCC bit is cleared to 0 and the WRE bit is set to 1 in TWCR.

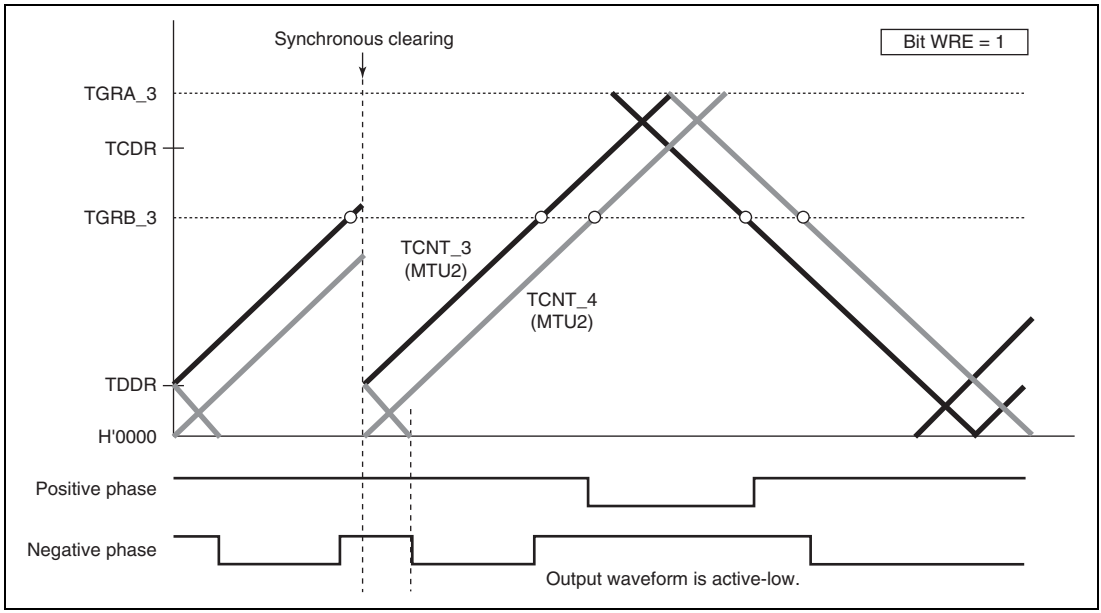



**Figure 10.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 10.56; Bit WRE of TWCR in MTU2 is 1)**

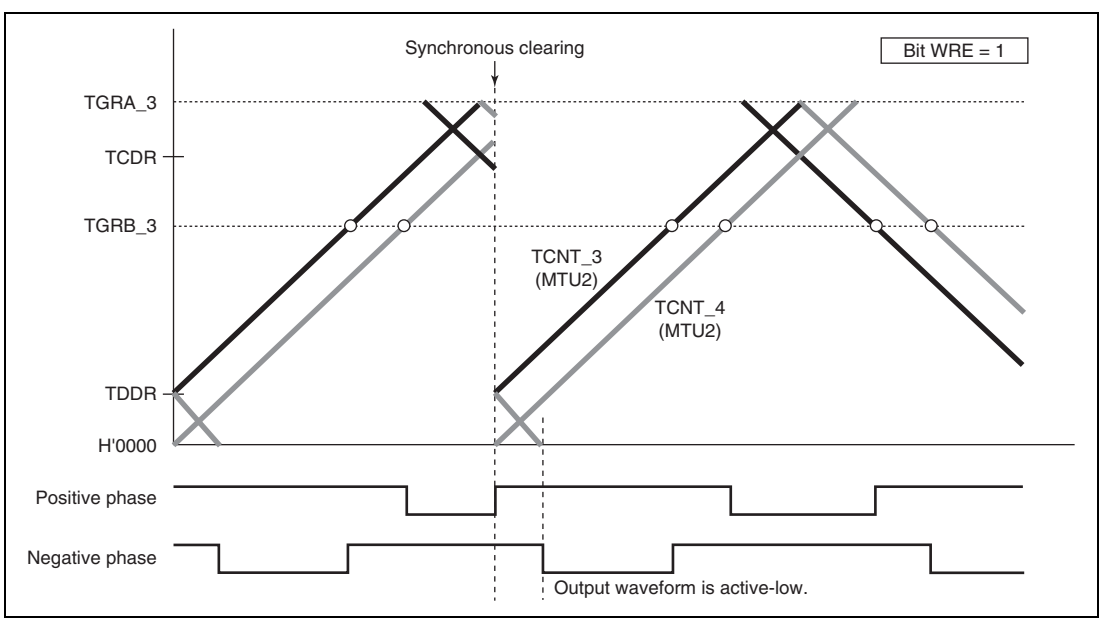



**Figure 10.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 10.56; Bit WRE of TWCR in MTU2 is 1)**

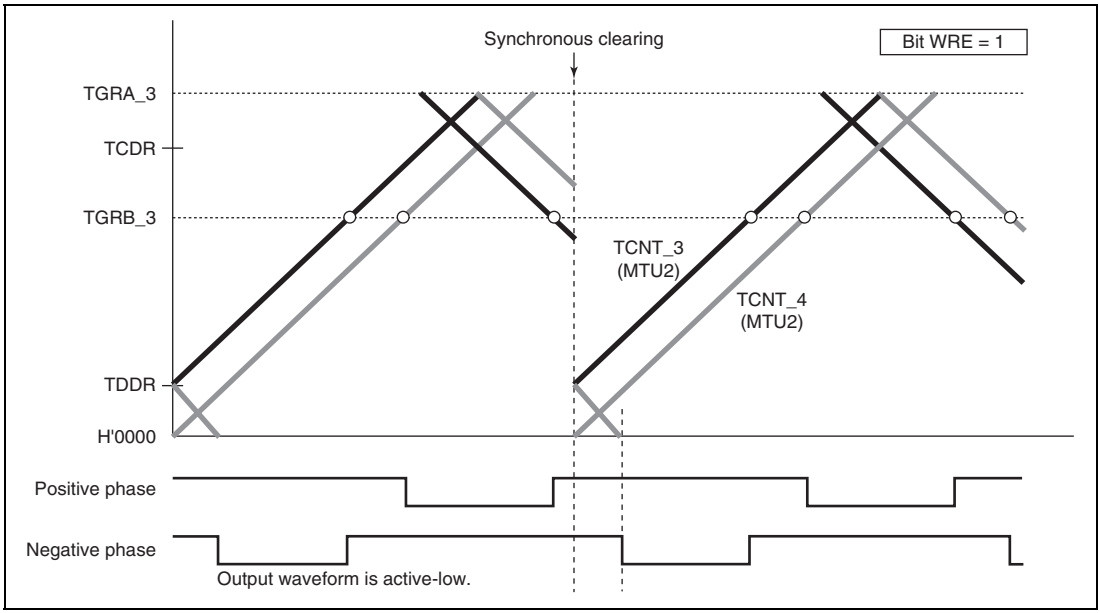



**Figure 10.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 10.56; Bit WRE of TWCR is 1)**

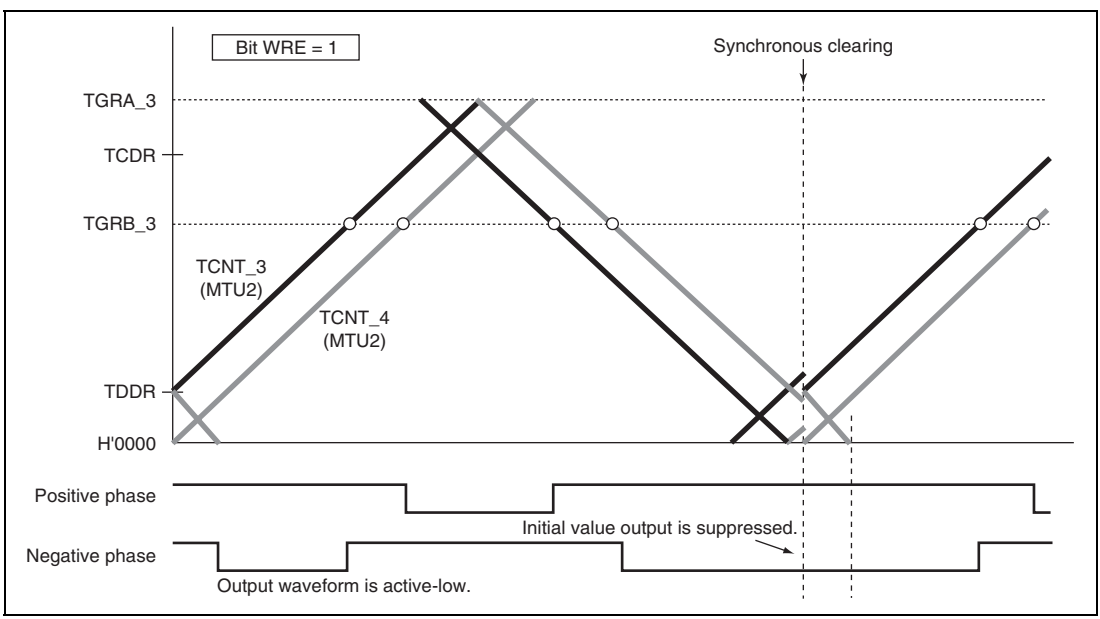



**Figure 10.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 10.56; Bit WRE of TWCR is 1)**

15. Suppressing MTU2–MTU2S Synchronous Counter Clearing

In the MTU2S, setting the SCC bit in TWCR to 1 suppresses synchronous counter clearing caused by the MTU2.

Synchronous counter clearing is suppressed only within the interval shown in figure 10.62. When using this function, the MTU2S should be set to complementary PWM mode.

For details of synchronous clearing caused by the MTU2, refer to the description about MTU2S counter clearing caused by MTU2 flag setting source (MTU2-MTU2S synchronous counter clearing) in section 10.4.10, MTU2–MTU2S Synchronous Operation.

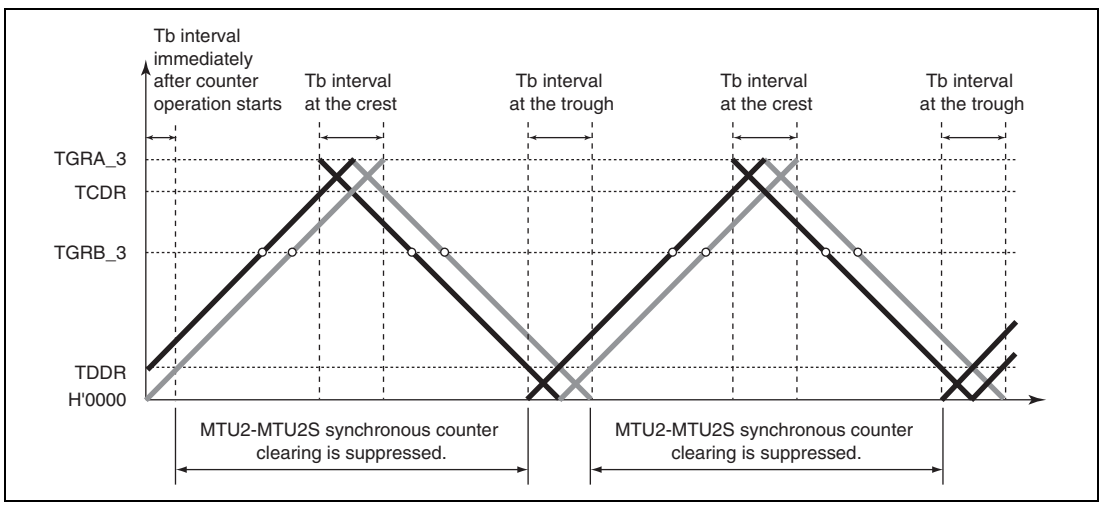



**Figure 10.62 MTU2–MTU2S Synchronous Clearing-Suppressed Interval Specified by SCC Bit in TWCR**

— Example of Procedure for Suppressing MTU2–MTU2S Synchronous Counter Clearing

An example of the procedure for suppressing MTU2–MTU2S synchronous counter clearing is shown in figure 10.63.

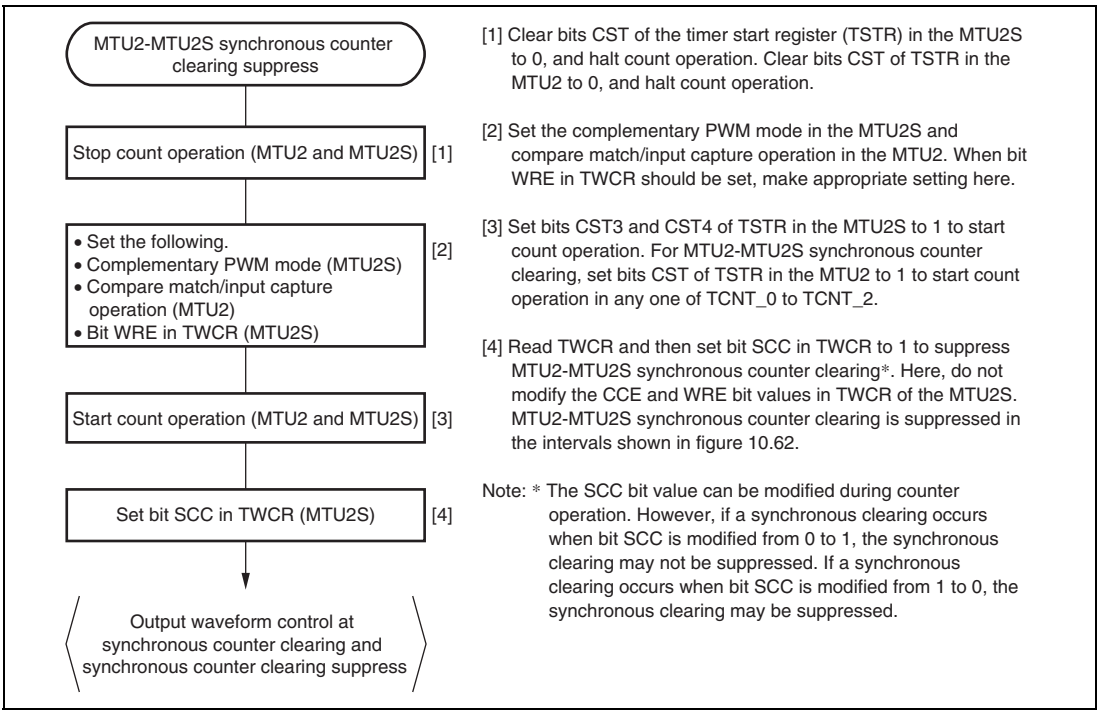



**Figure 10.63 Example of Procedure for Suppressing MTU2–MTU2S Synchronous Counter Clearing**

— Examples of Suppression of MTU2–MTU2S Synchronous Counter Clearing

Figures 10.64 to 10.67 show examples of operation in which the MTU2S operates in complementary PWM mode and MTU2–MTU2S synchronous counter clearing is suppressed by setting the SCC bit in TWCR in the MTU2S to 1. In the examples shown in figures 10.64 to 10.67, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 10.56, respectively.

In these examples, the WRE bit in TWCR of the MTU2S is set to 1.

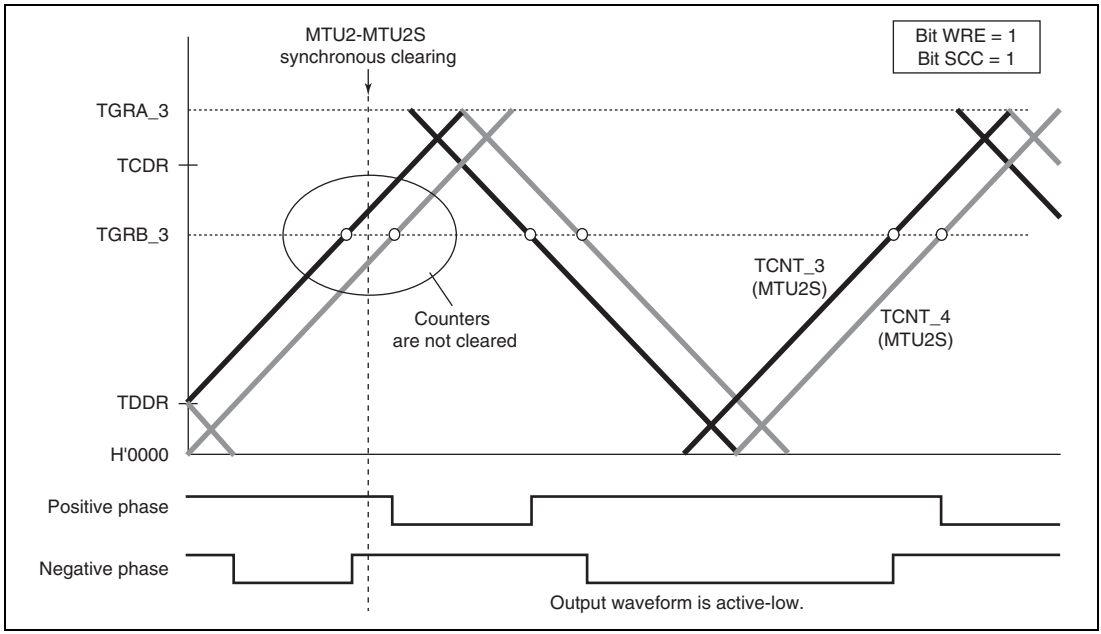



**Figure 10.64 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)**

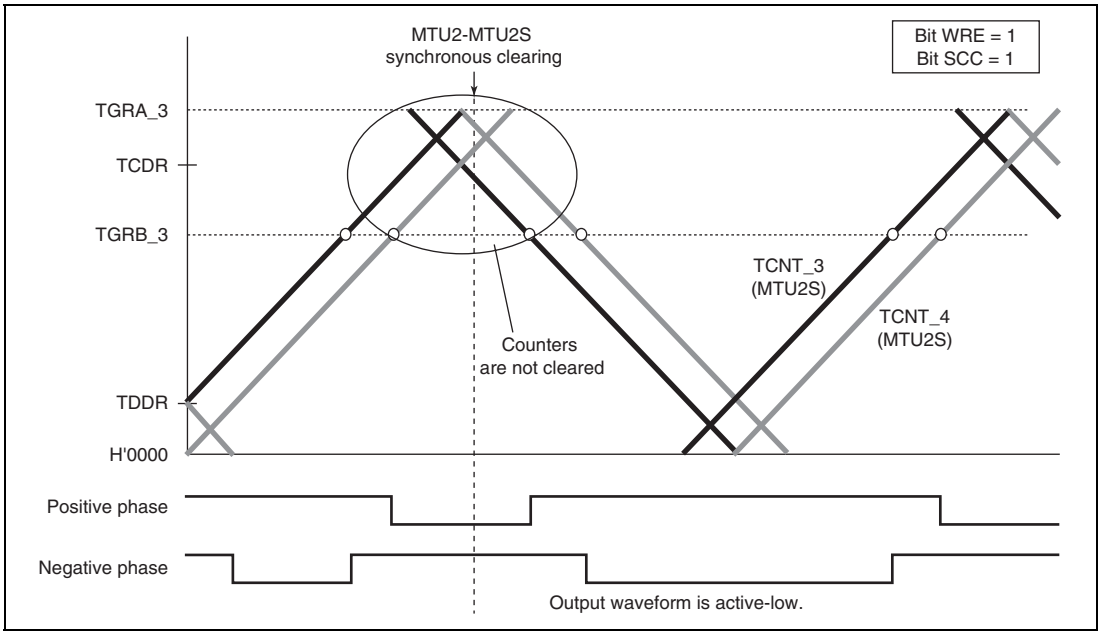



**Figure 10.65 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)**

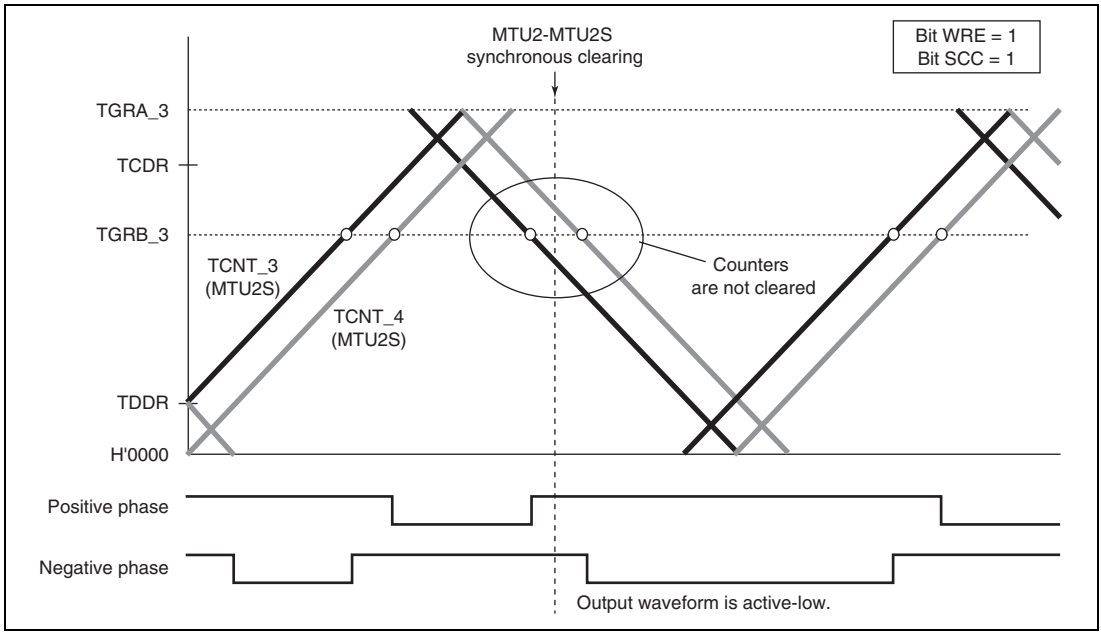



**Figure 10.66 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)**

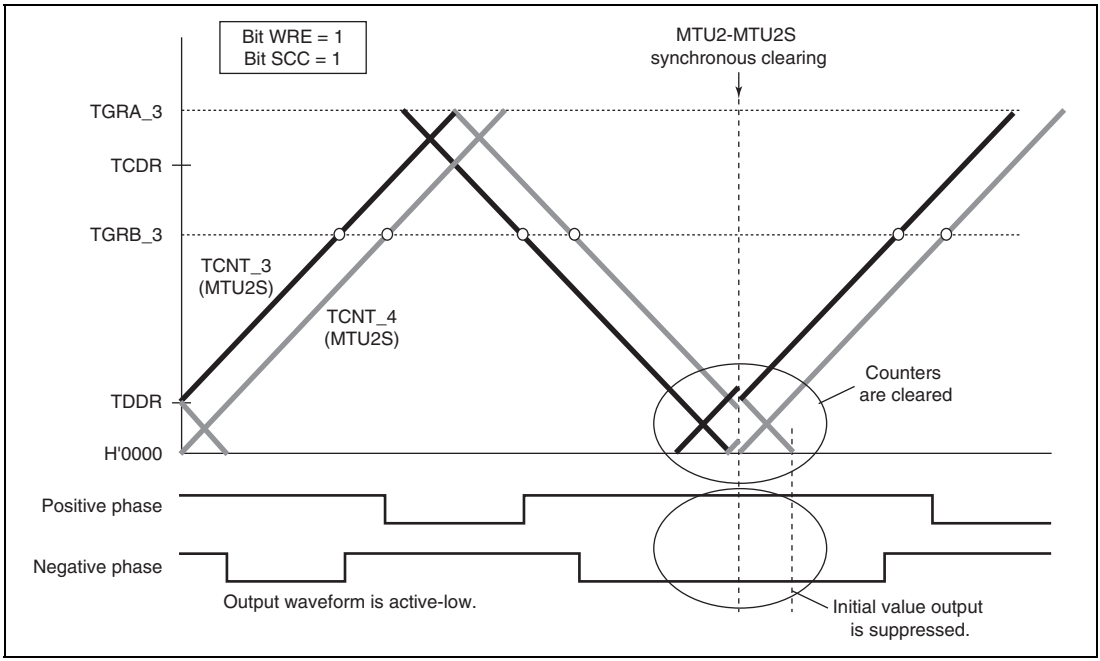



**Figure 10.67 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)**

16. Counter Clearing by TGRA_3 Compare Match

In complementary PWM mode, by setting the CCE bit in the timer waveform control register (TWCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by TGRA_3 compare match.

Figure 10.68 illustrates an operation example.

Notes: 1. Use this function only in complementary PWM mode 1 (transfer at crest)
2. Do not specify synchronous clearing by another channel (do not set the SYNC0 to SYNC4 bits in the timer synchronous register (TSYR) to 1 or the CE0A, CE0B, CE0C, CE0D, CE1A, CE1B, CE1C, and CE1D bits in the timer synchronous clear register (TSYCR) to 1).
3. Do not set the PWM duty value to H'0000.
4. Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

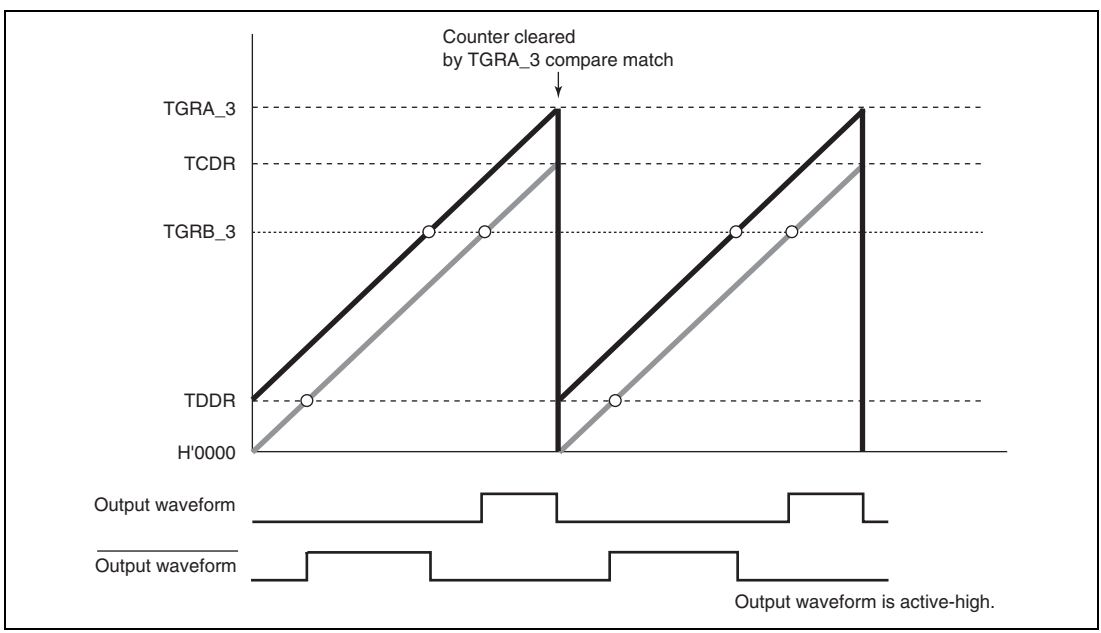



**Figure 10.68 Example of Counter Clearing Operation by TGRA_3 Compare Match**

17. Example of AC Synchronous Motor (Brushless DC Motor) Drive Waveform Output

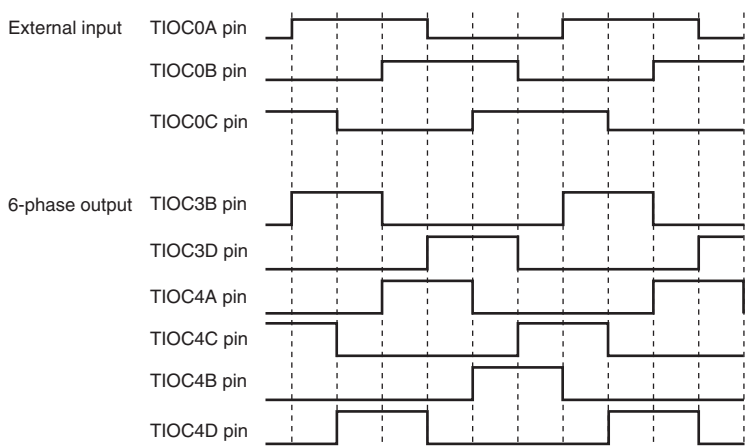
In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figures 10.69 to 10.72 show examples of brushless DC motor drive waveforms created using TGCR.

When output phase switching for a 3-phase brushless DC motor is performed by means of external signals detected with a Hall element, etc., clear the FB bit in TGCR to 0. In this case, the external signals indicating the polarity position are input to channel 0 timer input pins TIOC0A, TIOC0B, and TIOC0C (set with PFC). When an edge is detected at pin TIOC0A, TIOC0B, or TIOC0C, the output on/off state is switched automatically.

When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1.

The drive waveforms are output from the complementary PWM mode 6-phase output pins. With this 6-phase output, in the case of on output, it is possible to use complementary PWM mode output and perform chopping output by setting the N bit or P bit to 1. When the N bit or P bit is 0, level output is selected.

The 6-phase output active level (on output level) can be set with the OLSN and OLSP bits in the timer output control register (TOCR) regardless of the setting of the N and P bits.




**When BDC = 1, N = 0, P = 0, FB = 0, output active level = high**

**Figure 10.69 Example of Output Phase Switching by External Input (1)**

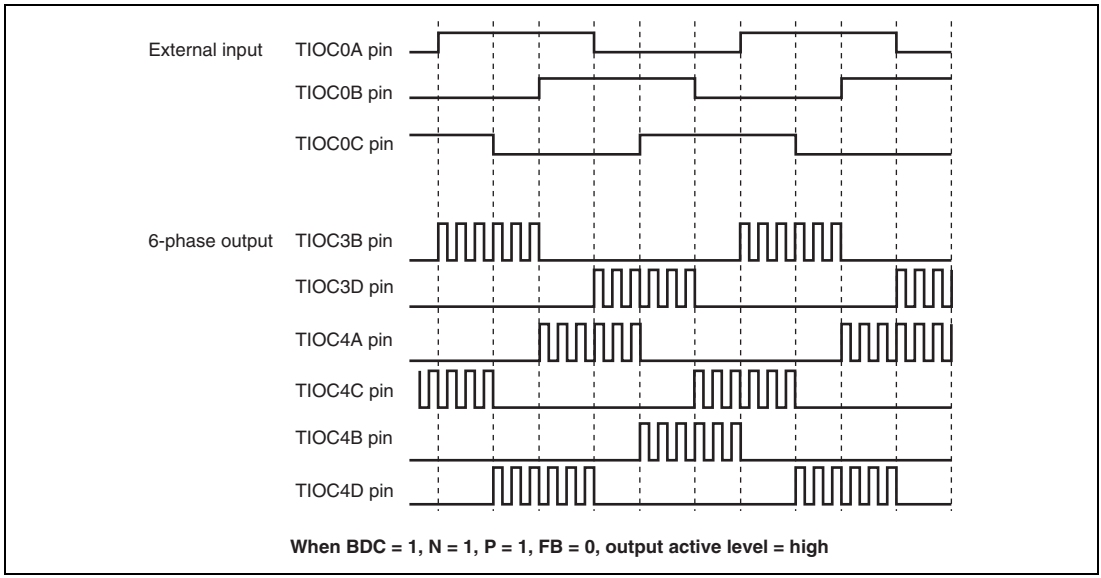



**Figure 10.70 Example of Output Phase Switching by External Input (2)**

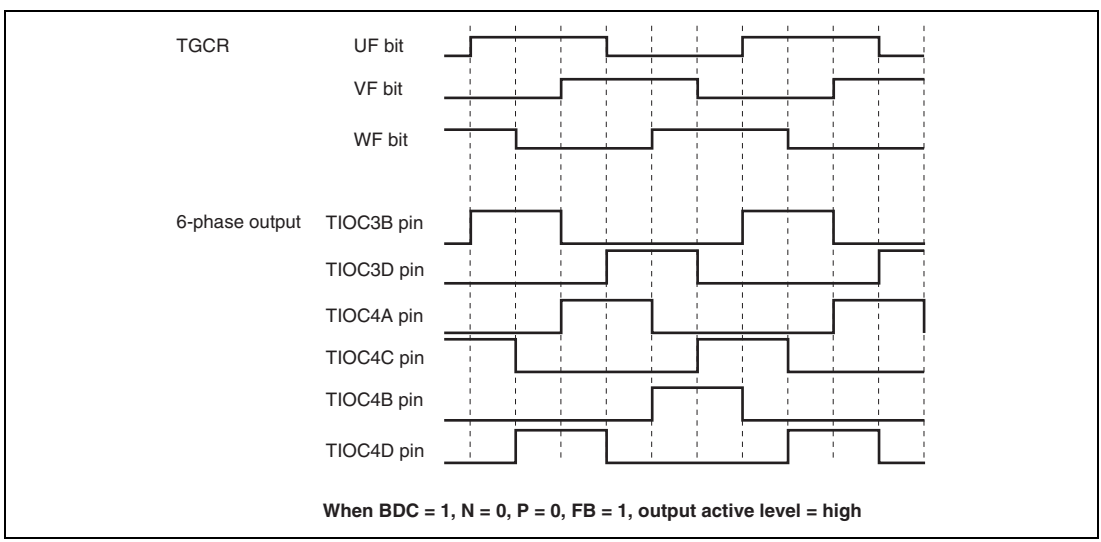



**Figure 10.71 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)**

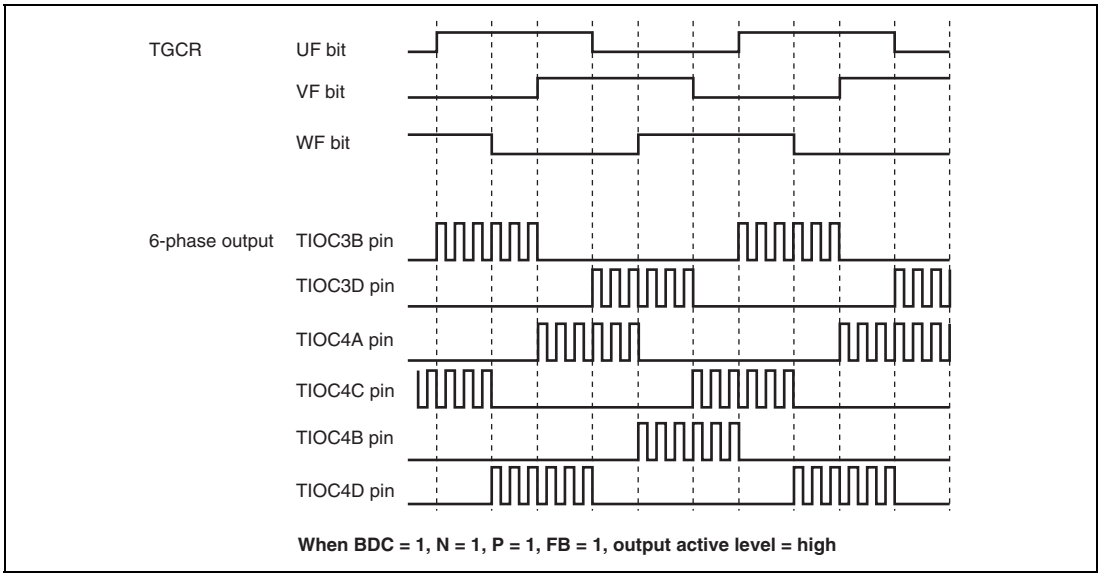



**Figure 10.72 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)**

18. 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.

**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).

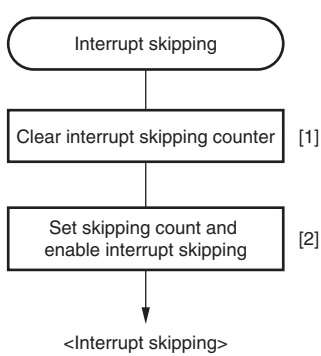
Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the timer buffer transfer register (TBTER). For the linkage with buffer registers, refer to description 3, Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the timer A/D converter request control register (TADCR). For the linkage with the A/D converter start request delaying function, refer to section 10.4.9, A/D Converter Start Request Delaying Function.

The setting of the timer interrupt skipping setting register (TITCR) must be done while the TGIA_3 and TCIV_4 interrupt requests are disabled by the settings of 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.

1. Example of Interrupt Skipping Operation Setting Procedure

   Figure 10.73 shows an example of the interrupt skipping operation setting procedure. Figure 10.74 shows the periods during which interrupt skipping count can be changed.




[1] Set bits T3AEN and T4VEN in the timer interrupt skipping set register (TITCR) to 0 to clear the skipping counter.

[2] Specify the interrupt skipping count within the range from 0 to 7 times in bits 3ACOR2 to 3ACOR0 and 4VCOR2 to 4VCOR0 in TITCR, and enable interrupt skipping through bits T3AEN and T4VEN.

Note: The setting of TITCR must be done while the TGIA_3 and TCIV_4 interrupt requests are disabled by the settings of TIER_3 and TIER_4 along with under the conditions in which TGFA_3 and TCFV_4 flag settings by compare match never occur.
Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter.

**Figure 10.73 Example of Interrupt Skipping Operation Setting Procedure**

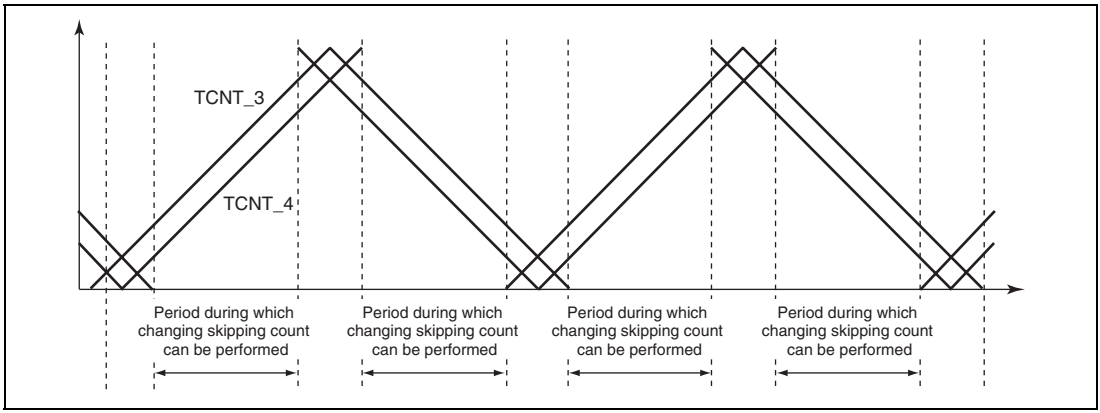



**Figure 10.74 Periods during which Interrupt Skipping Count can be Changed**

2. Example of Interrupt Skipping Operation

   Figure 10.75 shows an example of TGIA_3 interrupt skipping in which the interrupt skipping count is set to three by the 3ACOR bit and the T3AEN bit is set to 1 in the timer interrupt skipping set register (TITCR).

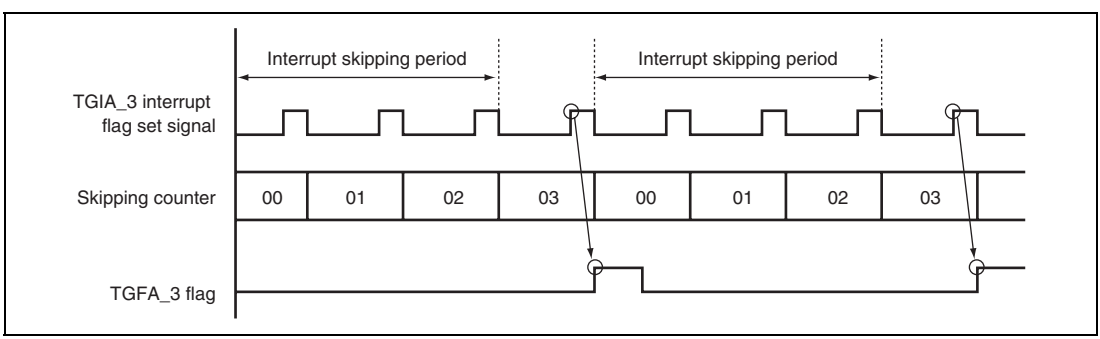



**Figure 10.75 Example of Interrupt Skipping Operation**

3. Buffer Transfer Control Linked with Interrupt Skipping

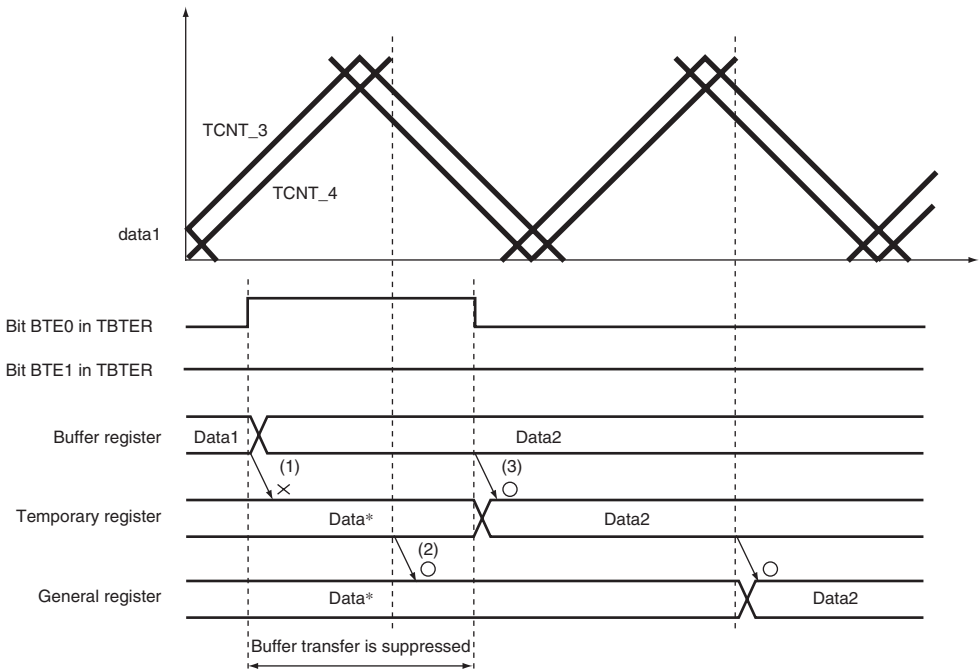
   In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

   Figure 10.76 shows an example of operation when buffer transfer is suppressed (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

   Figure 10.77 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE1 = 1 and BET0 = 0). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period.

   Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 10.78 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

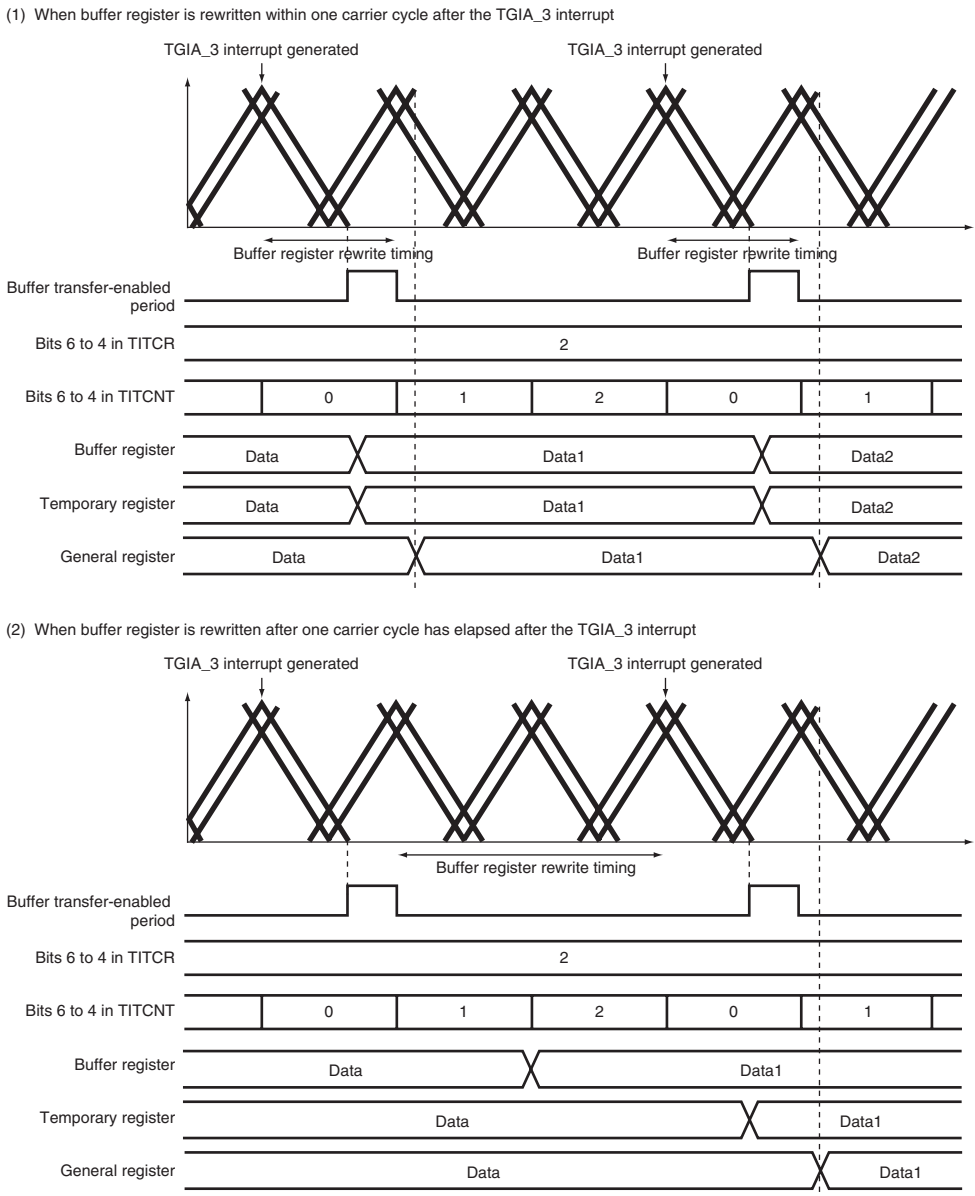



[Legend]

(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).
(2) Data is transferred from the temporary register to the general register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

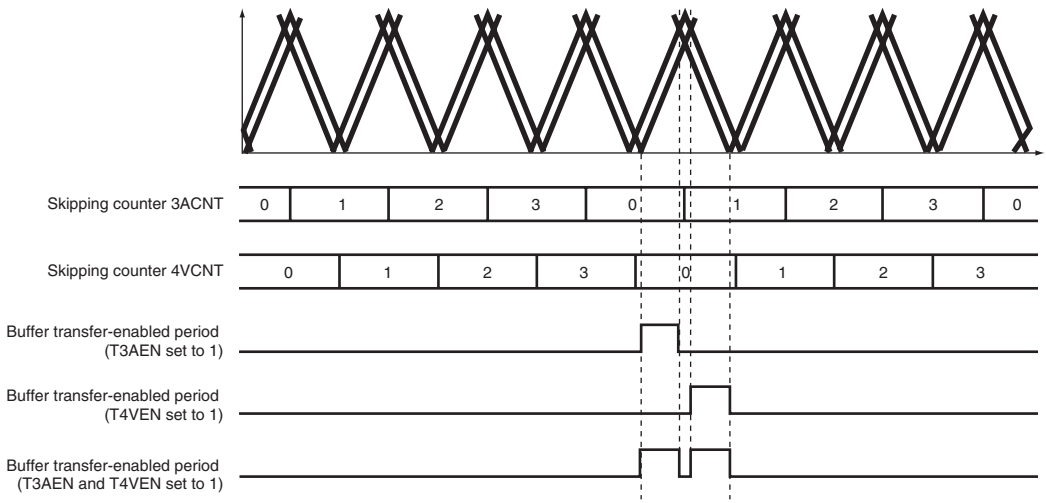
Note: * When buffer transfer at the crest is selected.

**Figure 10.76 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)**




Note: Bits MD3 to MD0 in TMDR_3 are set to 1101, selecting buffer transfer at the crest. The skipping count is set to two. T3AEN is set to 1, and T4VEN is cleared to 0.

**Figure 10.77 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)**




Note: Bits MD3 to MD0 in TMDR_3 are set to 1111, selecting buffer transfer at the crest and trough. The skipping count is set to three. T3AEN and T4VEN are set to 1.

**Figure 10.78 Relationship between Bits T3AEN and T4VEN in Timer Interrupt Skipping Set Register (TITCR) and Buffer Transfer-Enabled Period**

**Complementary PWM Mode Output Protection Function:**

Complementary PWM mode output has the following protection functions.

1. 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.

2. Halting of PWM output by external signal

   The 6-phase PWM output pins can be set automatically to the high-impedance state by inputting specified external signals. There are four external signal input pins.

   See section 12, Port Output Enable (POE), for details.

3. Halting of PWM output when oscillator is stopped

   If it is detected that the clock input to this LSI has stopped, the 6-phase PWM output pins automatically go to the high-impedance state. The pin states are not guaranteed when the clock is restarted.

   See section 4.7, Function for Detecting Oscillator Stop.

### 10.4.9 A/D Converter Start Request Delaying Function

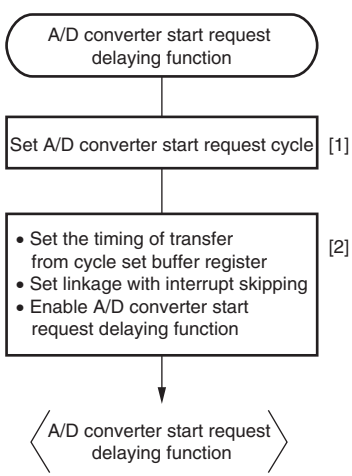
A/D converter start requests can be issued in channel 4 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4), and timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4).

The A/D converter start request delaying function compares TCNT_4 with TADCORA_4 or TADCORB_4, and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN).

A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in TADCR.

1. Example of Procedure for Specifying A/D Converter Start Request Delaying Function

   Figure 10.79 shows an example of procedure for specifying the A/D converter start request delaying function.




[1] Set the cycle in the timer A/D converter start request cycle buffer register (TADCOBRA_4 or TADCOBRB_4) and timer A/D converter start request cycle register (TADCORA_4 or TADCORB_4). (The same initial value must be specified in the cycle buffer register and cycle register.)

[2] Use bits BF1 and BF2 in the timer A/D converter start request control register (TADCR) to specify the timing of transfer from the timer A/D converter start request cycle buffer register to A/D converter start request cycle register.
- Specify whether to link with interrupt skipping through bits ITA3AE, ITA4VE, ITB3AE, and ITB4VE.
- Use bits TU4AE, DT4AE, UT4BE, and DT4BE to enable A/D conversion start requests (TRG4AN or TRG4BN).

Notes: 1. Perform TADCR setting while TCNT_4 is stopped.
2. Do not set BF1 to 1 when complementary PWM mode is not selected.
3. Do not set ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, or DT4BE to 1 when complementary PWM mode is not selected.

**Figure 10.79 Example of Procedure for Specifying A/D Converter Start Request Delaying Function**

2. Basic Operation Example of A/D Converter Start Request Delaying Function

   Figure 10.80 shows a basic example of A/D converter request signal (TRG4AN) operation when the trough of TCNT_4 is specified for the buffer transfer timing and an A/D converter start request signal is output during TCNT_4 down-counting.

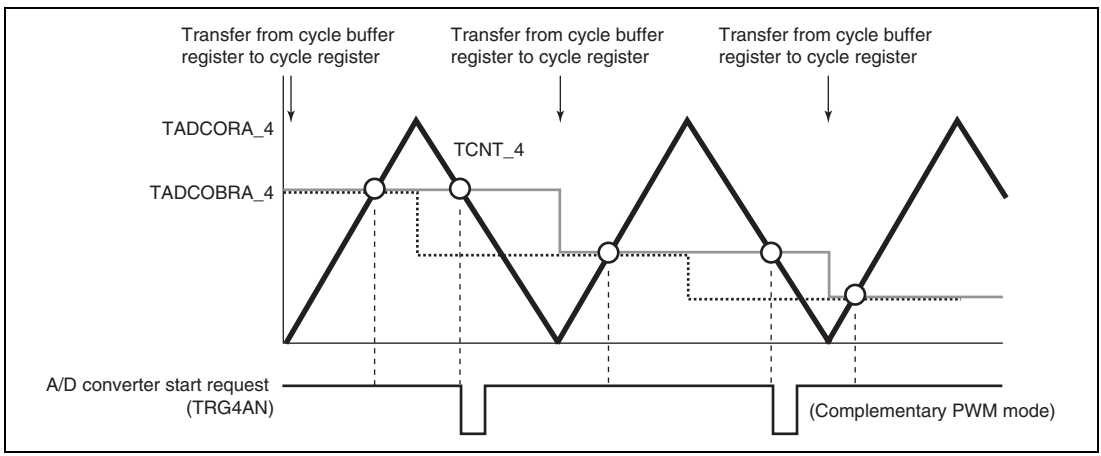



**Figure 10.80 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation**

3. 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).

4. A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

   A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR).

   Figure 10.81 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

   Figure 10.82 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and A/D converter start requests are linked with interrupt skipping.

Note: This function must be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).

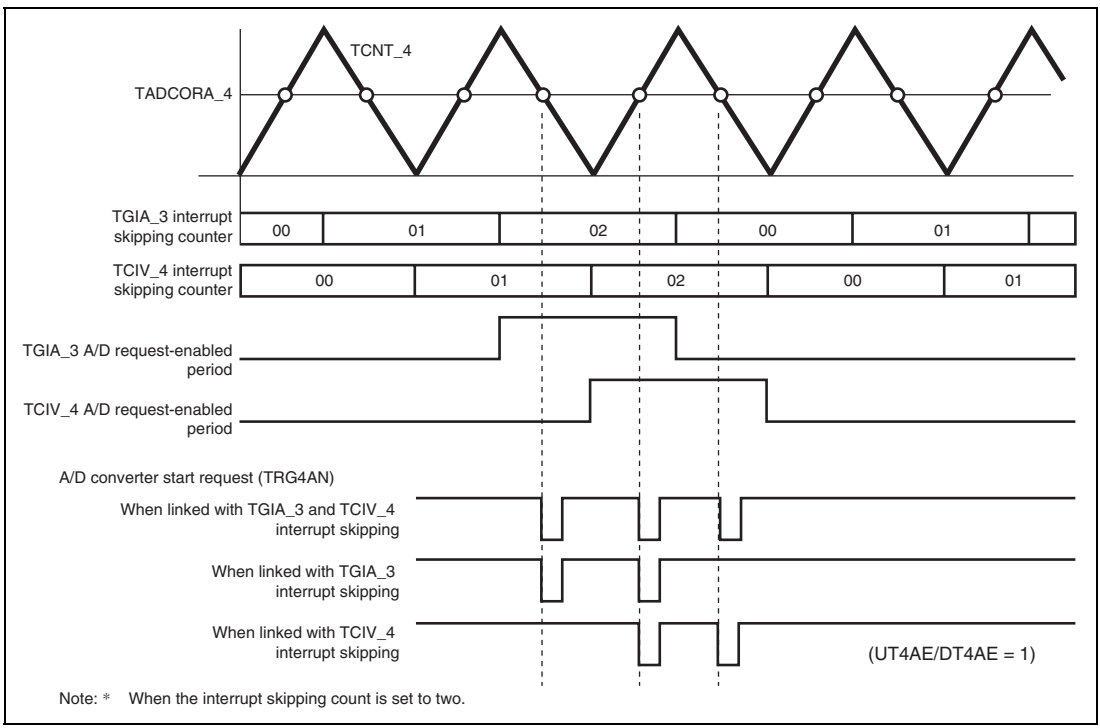



**Figure 10.81 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping**

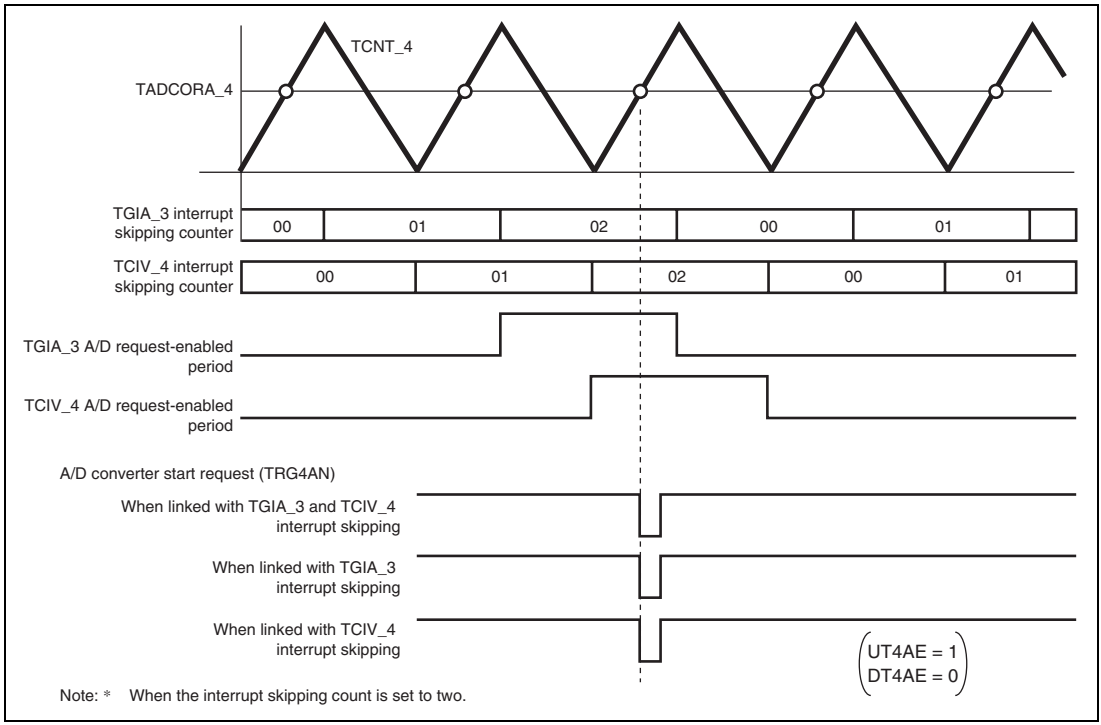
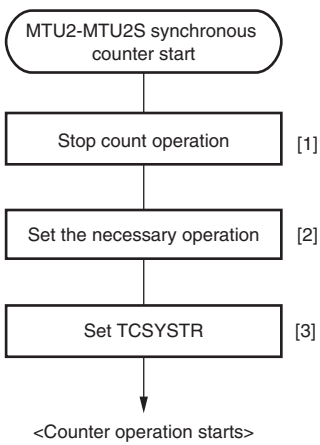



**Figure 10.82 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping**

### 10.4.10 MTU2–MTU2S Synchronous Operation

**MTU2–MTU2S Synchronous Counter Start:** The counters in the MTU2 and MTU2S which operate at different clock systems can be started synchronously by making the TCSYSTR settings in the MTU2.

1. Example of MTU2–MTU2S Synchronous Counter Start Setting Procedure
   Figure 10.83 shows an example of synchronous counter start setting procedure.




[1] Use TSTR registers in the MTU2 and MTU2S and halt the counters used for synchronous start operation.

[2] Specify necessary operation with appropriate registers such as TCR and TMDR.

[3] In TCSYSTR in the MTU2, set the bits corresponding to the counters to be started synchronously to 1. The TSTRs are automatically set appropriately and the counters start synchronously.

Notes: 1. Even if a bit in TCSYSTR corresponding to an operating counter is cleared to 0, the counter will not stop. To stop the counter, clear the corresponding bit in TSTR to 0 directly.
2. To start channels 3 and 4 in reset-synchronized PWM mode or complementary PWM mode, make appropriate settings in TCYSTR according to the TSTR setting for the respective mode. For details, refer to section 10.4.7, Reset-Synchronized PWM Mode, and section 10.4.8, Complementary PWM Mode.

**Figure 10.83 Example of Synchronous Counter Start Setting Procedure**

2. Examples of Synchronous Counter Start Operation

   Figures 10.84 (1), 10.84 (2), 10.84 (3), and 10.84 (4) show examples of synchronous counter start operation when the clock frequency ratio between the MTU2 and MTU2S is 1:1, 1:2, 1:3, and 1:4, respectively.

   In these examples, the counter clock of the MTU2 is MPϕ/1.

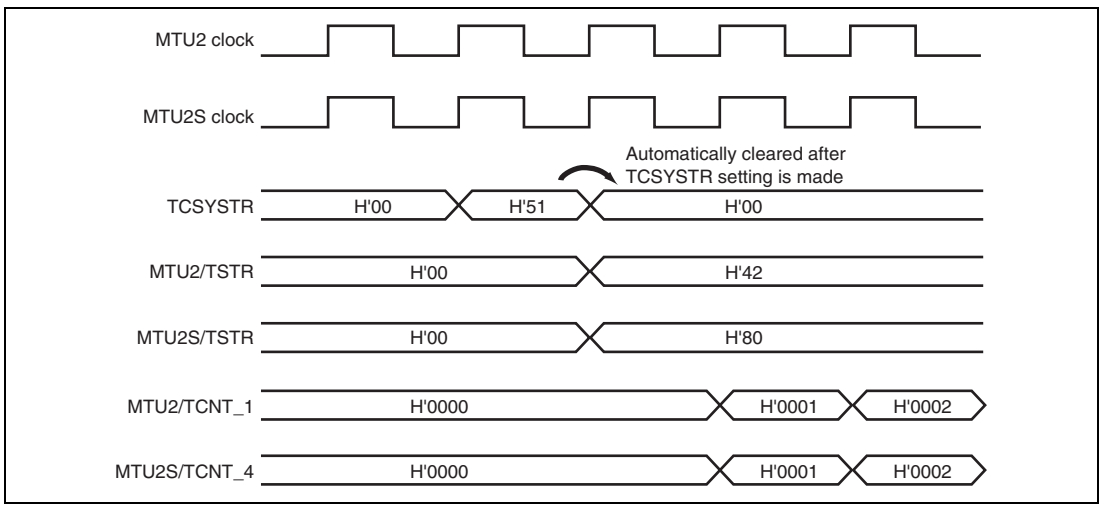



**Figure 10.84 (1) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:1)**

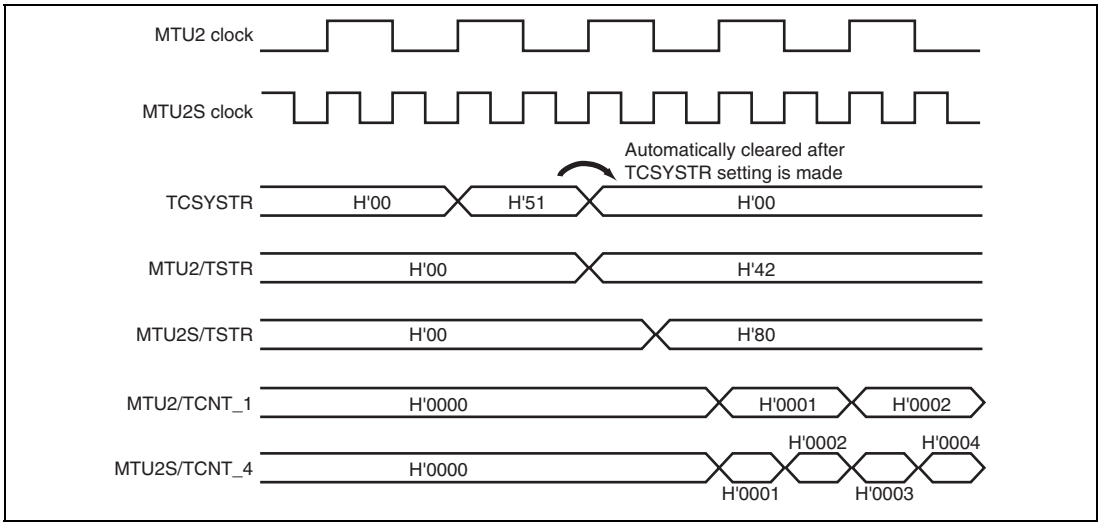



**Figure 10.84 (2) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:2)**

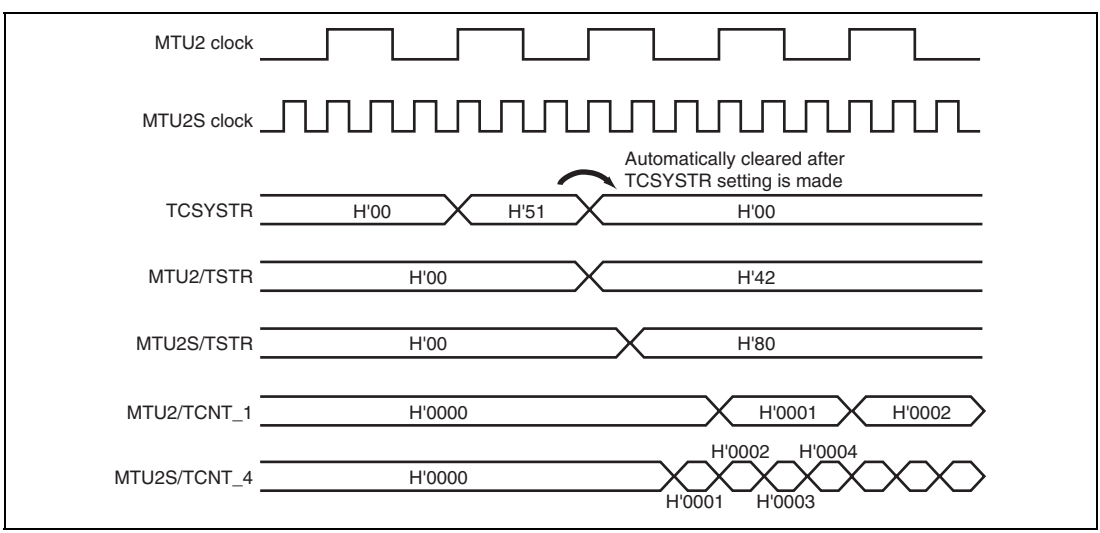



**Figure 10.84 (3) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:3)**

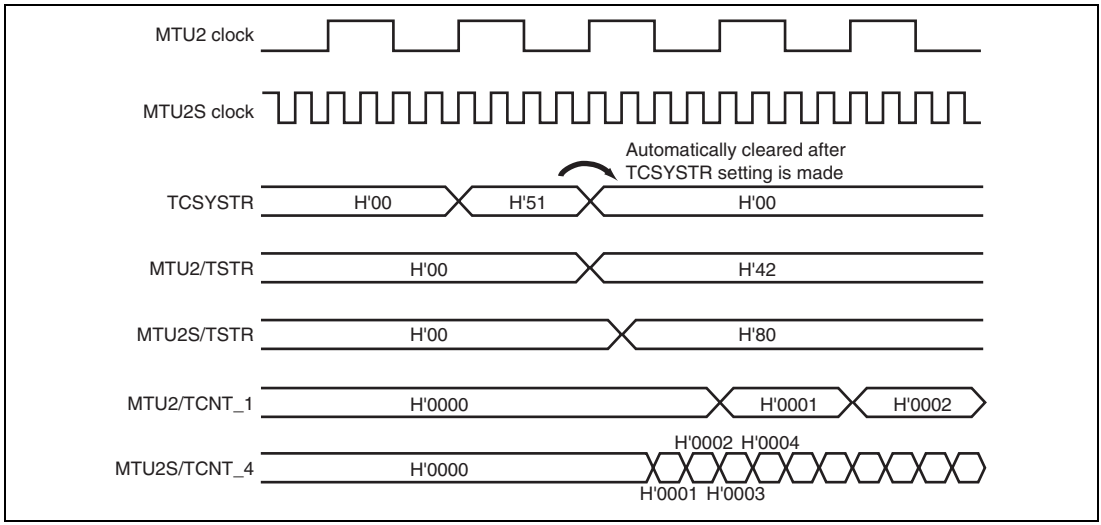



**Figure 10.84 (4) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:4)**

**MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (MTU2–MTU2S Synchronous Counter Clearing):** The MTU2S counters can be cleared by sources for setting the flags in TSR_0 to TSR_2 in the MTU2 through the TSYCR_3 settings in the MTU2S.

1. Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source

   Figure 10.85 shows an example of procedure for specifying MTU2S counter clearing by MTU2 flag setting source.

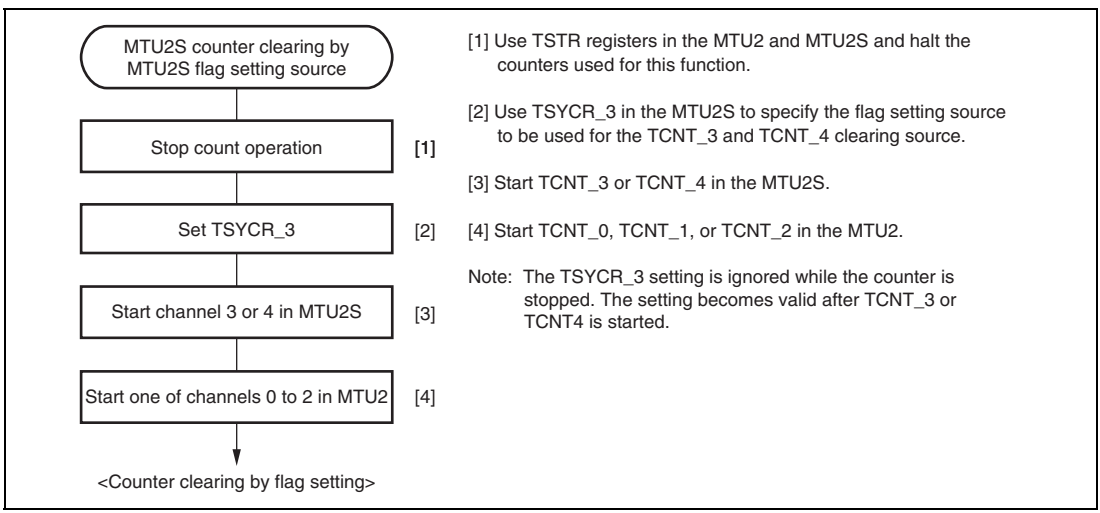



**Figure 10.85 Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source**

2. Examples of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source

   Figures 10.86 (1) and 10.86 (2) show examples of MTS2S counter clearing caused by MTU2 flag setting source.

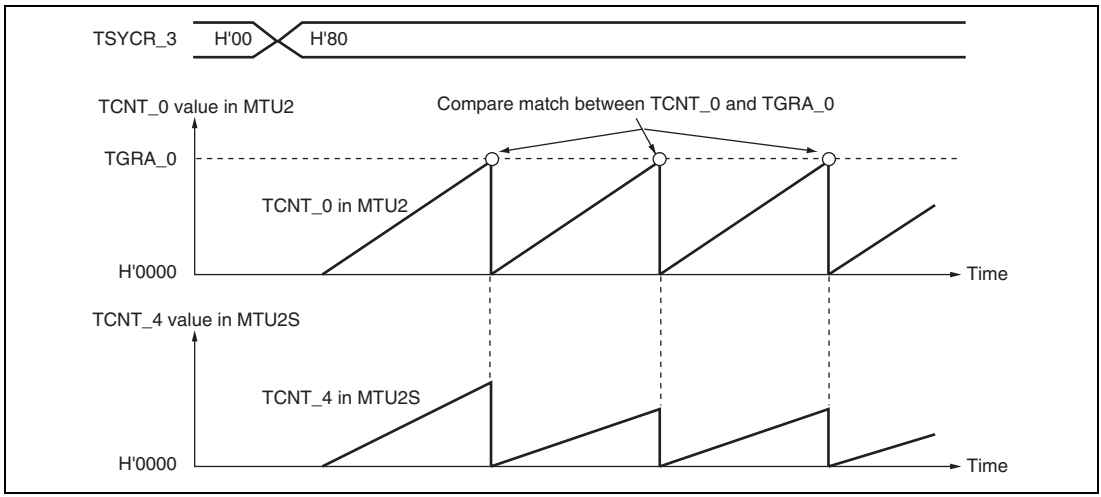



**Figure 10.86 (1) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (1)**

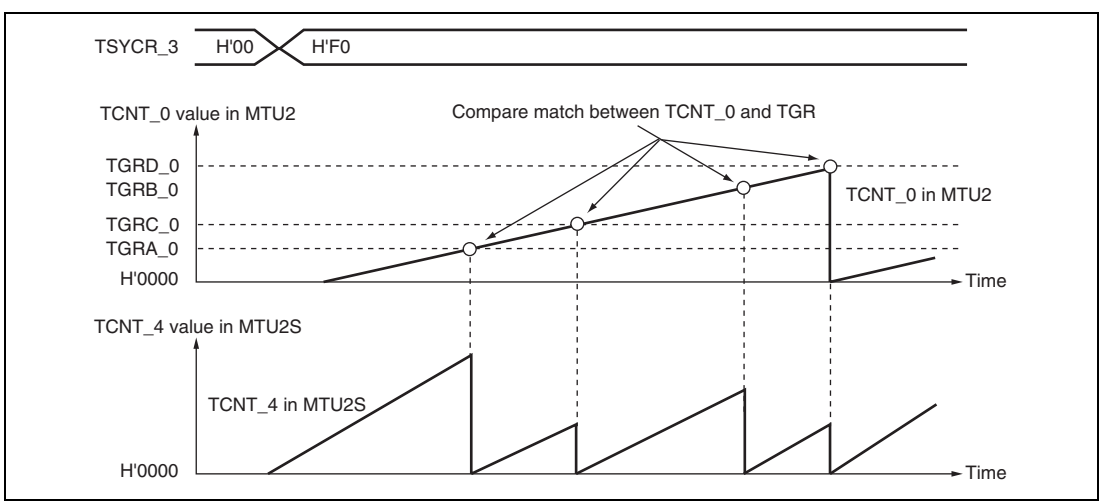



**Figure 10.86 (2) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (2)**

### 10.4.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in channel 5.

**Example of External Pulse Width Measurement Setting Procedure:**

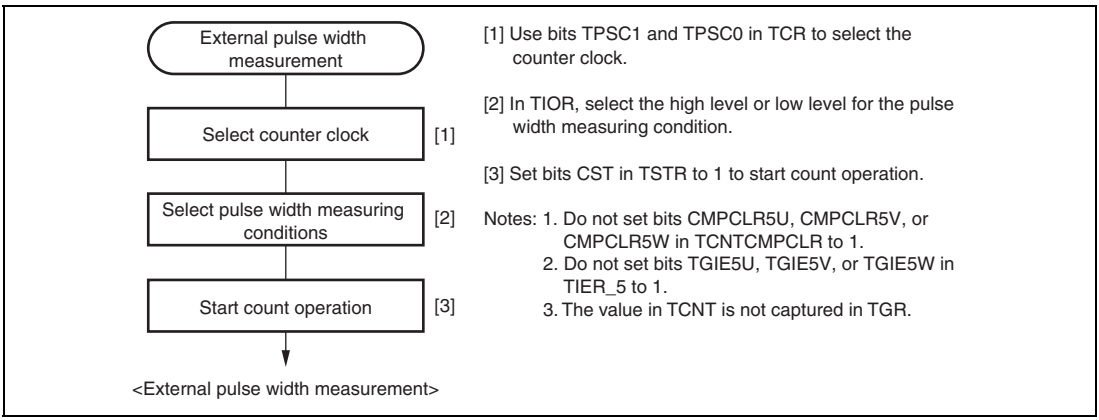



**Figure 10.87 Example of External Pulse Width Measurement Setting Procedure**

**Example of External Pulse Width Measurement:**

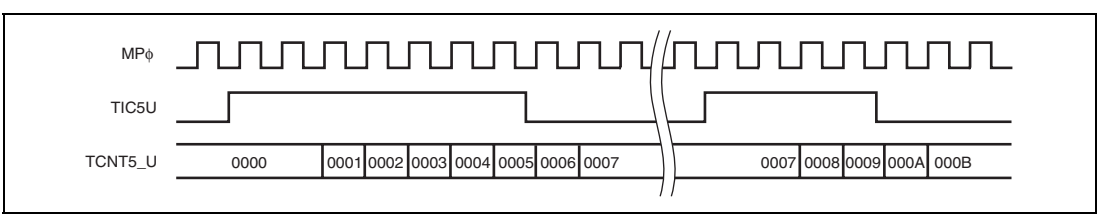



**Figure 10.88 Example of External Pulse Width Measurement (Measuring High Pulse Width)**

### 10.4.12 Dead Time Compensation

By measuring the delay of the output waveform and reflecting it to duty, the external pulse width measurement function can be used as the dead time compensation function while the complementary PWM is in operation.

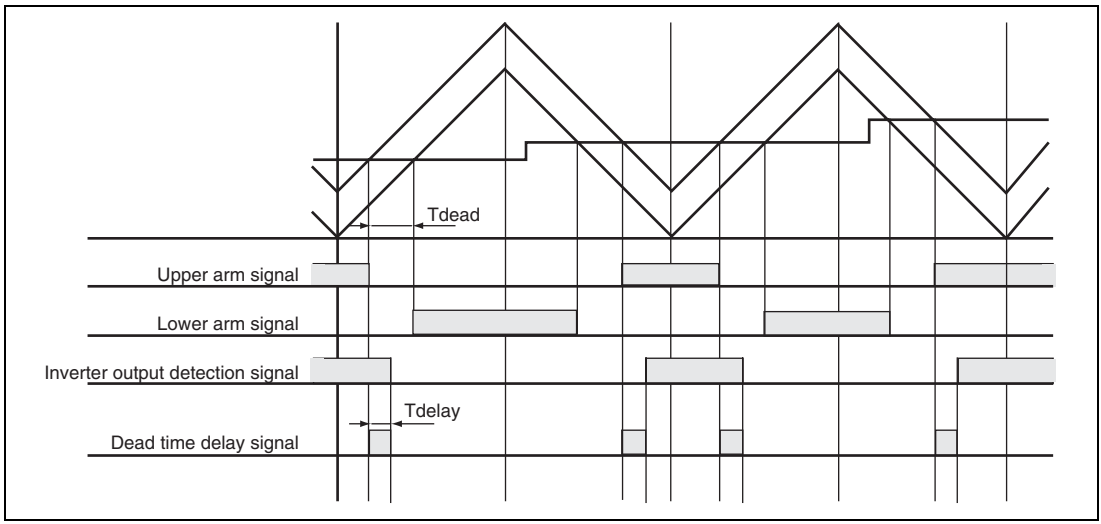
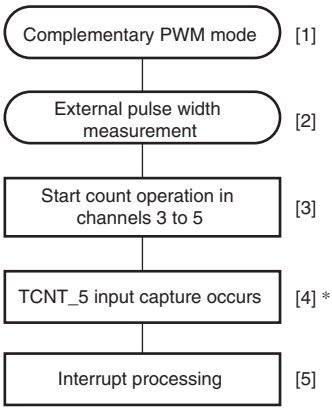



**Figure 10.89 Delay in Dead Time in Complementary PWM Operation**

**Example of Dead Time Compensation Setting Procedure:** Figure 10.90 shows an example of dead time compensation setting procedure by using three counters in channel 5.




[1] Place channels 3 and 4 in complementary PWM mode. For details, refer to section 10.4.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in channel 5. For details, refer to section 10.4.11, External Pulse Width Measurement.

[3] Set bits CST3 and CST4 in TSTR and bits CST5U, CST5V, and CST5W in TSTR2 to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the TCNT_5 value is captured in TGR_5.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA_3) or trough (TCIV_4) in complementary PWM mode, read the TGRU_5 value, calculate the difference in time in TGRB_3, and write the corrected value to TGRD_3 in the interrupt processing.
For the V phase and W phase, read the TGRV_5 and TGRW_5 values and write the corrected values to TGRC_4 and TGRD_4, respectively, in the same way as for U-phase compensation.
The TCNT_5 value should be cleared through the TCNTCMPCLR setting or by software.

Notes: The PFC settings must be completed in advance.
* As an interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in TIER_5.

**Figure 10.90 Example of Dead Time Compensation Setting Procedure**

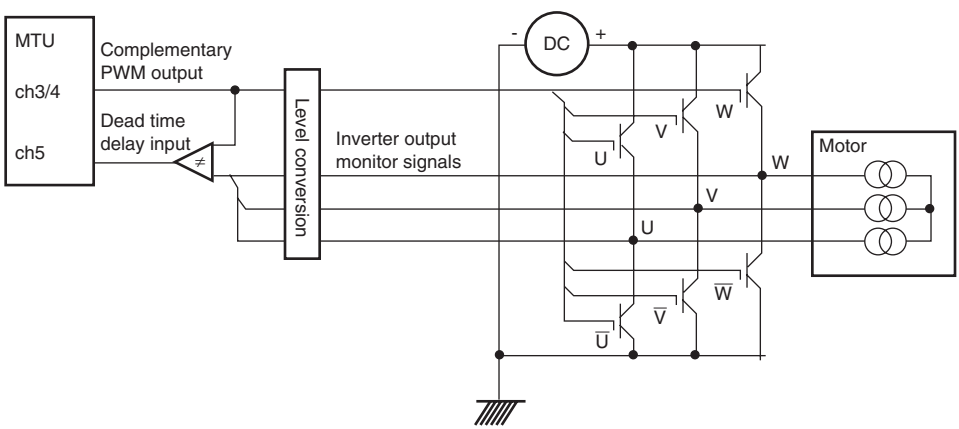



**Figure 10.91 Example of Motor Control Circuit Configuration**

### 10.4.13 TCNT Capture at Crest and/or Trough in Complementary PWM Operation

The TCNT value is captured in TGR at either the crest or trough or at both the crest and trough during complementary PWM operation. The timing for capturing in TGR can be selected by TIOR.

Figure 10.92 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).

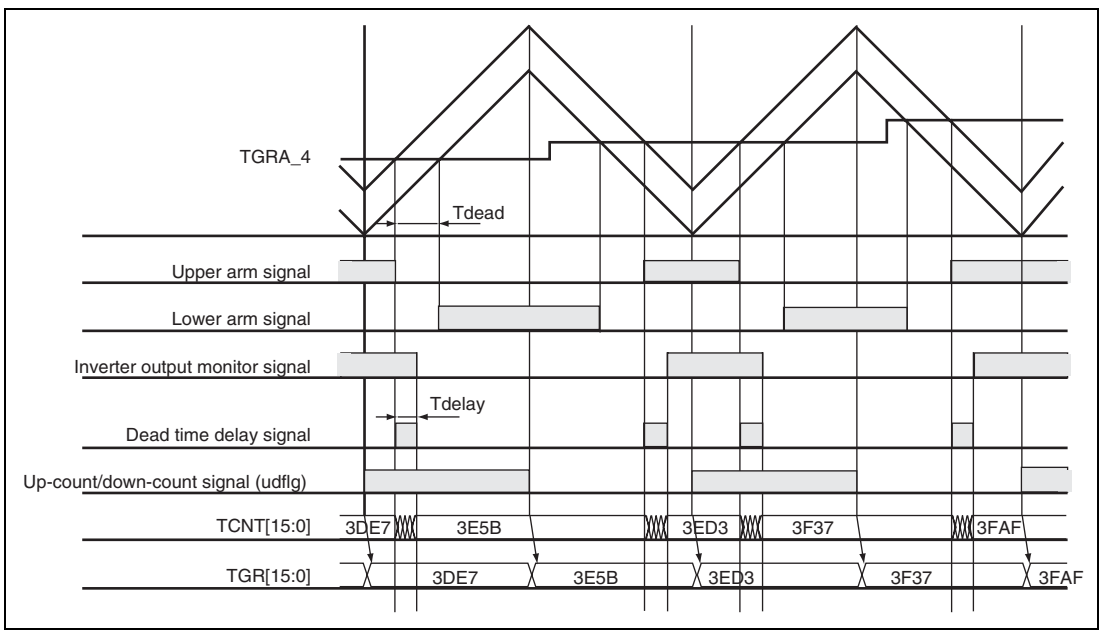



**Figure 10.92 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation**

## 10.5 Interrupt Sources

### 10.5.1 Interrupt Sources and Priorities

There are three kinds of MTU2 interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, however the priority order within a channel is fixed. For details, see section 6, Interrupt Controller (INTC).

Table 10.57 lists the MTU2 interrupt sources.

**Table 10.57 MTU2 Interrupts**

| Channel | Name | Interrupt Source | Interrupt Flag | DTC 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 | Possible | |
| | TGIC_0 | TGRC_0 input capture/compare match | TGFC_0 | Possible | |
| | TGID_0 | TGRD_0 input capture/compare match | TGFD_0 | 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 | 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 | Possible | |
| | TCIV_2 | TCNT_2 overflow | TCFV_2 | Not possible | |
| | TCIU_2 | TCNT_2 underflow | TCFU_2 | Not possible | |
| 3 | TGIA_3 | TGRA_3 input capture/compare match | TGFA_3 | Possible | |
| | TGIB_3 | TGRB_3 input capture/compare match | TGFB_3 | Possible | |
| | TGIC_3 | TGRC_3 input capture/compare match | TGFC_3 | Possible | |
| | TGID_3 | TGRD_3 input capture/compare match | TGFD_3 | 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 | Possible | |
| | TGIC_4 | TGRC_4 input capture/compare match | TGFC_4 | Possible | |
| | TGID_4 | TGRD_4 input capture/compare match | TGFD_4 | Possible | |
| | TCIV_4 | TCNT_4 overflow/underflow | TCFV_4 | Possible | |
| 5 | TGIU_5 | TGRU_5 input capture/compare match | TGFU_5 | Possible | |
| | TGIV_5 | TGRV_5 input capture/compare match | TGFV_5 | Possible | |
| | TGIW_5 | TGRW_5 input capture/compare match | TGFW_5 | Possible | Low |

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

**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.

**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.

**Underflow Interrupt:** An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The MTU2 has two underflow interrupts, one each for channels 1 and 2.

### 10.5.2 DTC Activation

**DTC Activation:** The DTC can be activated by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in channel 4. For details, see section 8, Data Transfer Controller (DTC).

A total of 20 MTU2 input capture/compare match interrupts and overflow interrupts can be used as DTC activation sources, four each for channels 0 and 3, two each for channels 1 and 2, five for channel 4, and three for channel 5.

### 10.5.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU2. Table 10.58 shows the relationship between interrupt sources and A/D converter start request signals.

**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.

**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.

**A/D Converter Activation by A/D Converter Start Request Delaying Function:** The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the TCNT_4 count matches the TADCORA or TADCORB value if the TAD4AE or TAD4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 10.4.9, A/D Converter Start Request Delaying Function.

A/D conversion will start if A/D converter start signal TRG4AN from the MTU2 is selected as the trigger in the A/D converter when TRG4AN is generated or if TRG4BN from the MTU2 is selected as the trigger in the A/D converter when TRG4BN is generated.

**Table 10.58 Interrupt Sources and A/D Converter Start Request Signals**

| Target Registers | Interrupt Source | A/D Converter Start Request Signal |
|---|---|---|
| TGRA_0 and TCNT_0 | Input capture/compare match | TRGAN |
| TGRA_1 and TCNT_1 | | |
| TGRA_2 and TCNT_2 | | |
| TGRA_3 and TCNT_3 | | |
| TGRA_4 and TCNT_4 | | |
| TCNT_4 | TCNT_4 Trough in complementary PWM mode | |
| TGRE_0 and TCNT_0 | Compare match | TRG0N |
| TADCORA and TCNT_4 | | TRG4AN |
| TADCORB and TCNT_4 | | TRG4BN |

## 10.6 Operation Timing

### 10.6.1 Input/Output Timing

**TCNT Count Timing:** Figures 10.93 and 10.94 show TCNT count timing in internal clock operation, and figure 10.95 shows TCNT count timing in external clock operation (normal mode), and figure 10.96 shows TCNT count timing in external clock operation (phase counting mode).

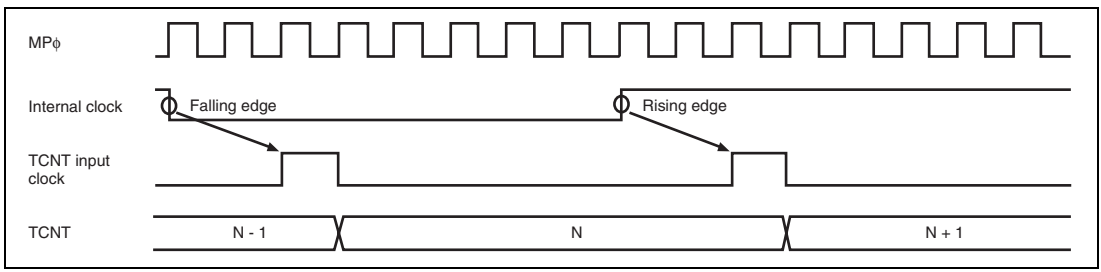



**Figure 10.93 Count Timing in Internal Clock Operation (Channels 0 to 4)**

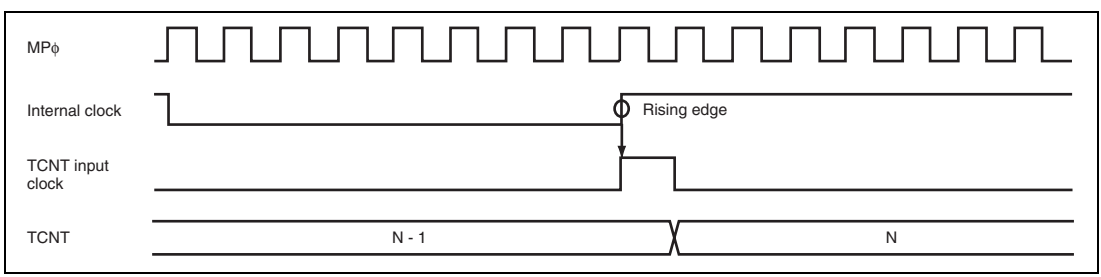



**Figure 10.94 Count Timing in Internal Clock Operation (Channel 5)**

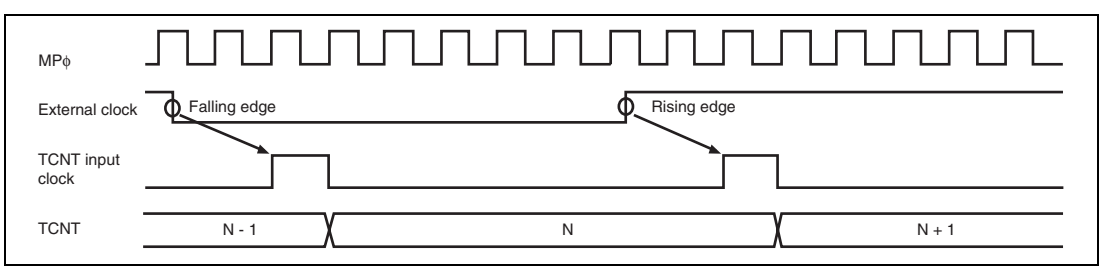



**Figure 10.95 Count Timing in External Clock Operation (Channels 0 to 4)**

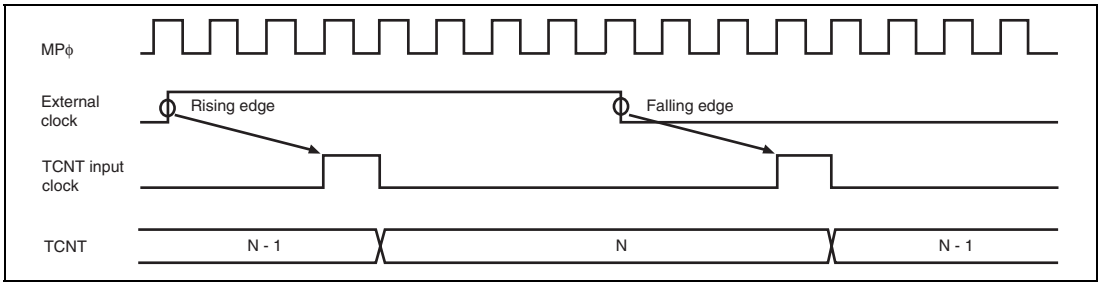



**Figure 10.96 Count Timing in External Clock Operation (Phase Counting Mode)**

**Output Compare Output Timing:** A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 10.97 shows output compare output timing (normal mode and PWM mode) and figure 10.98 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

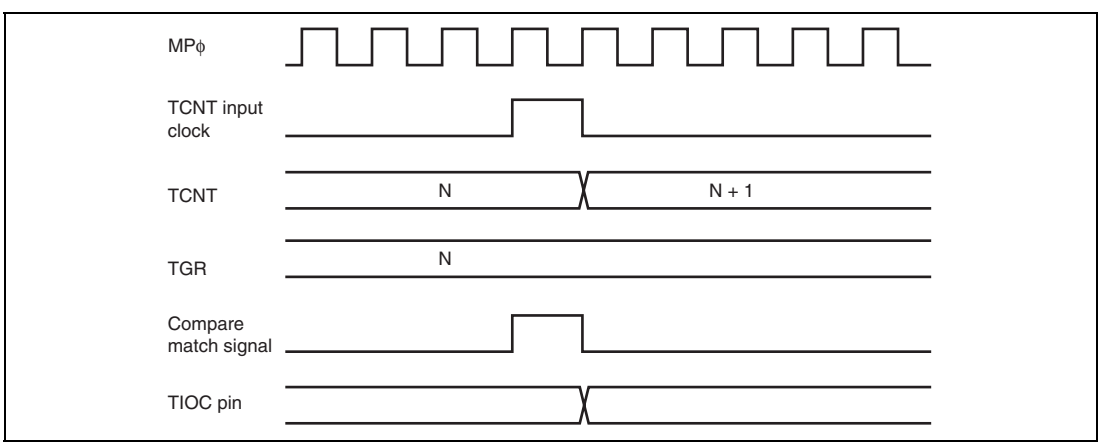



**Figure 10.97 Output Compare Output Timing (Normal Mode/PWM Mode)**

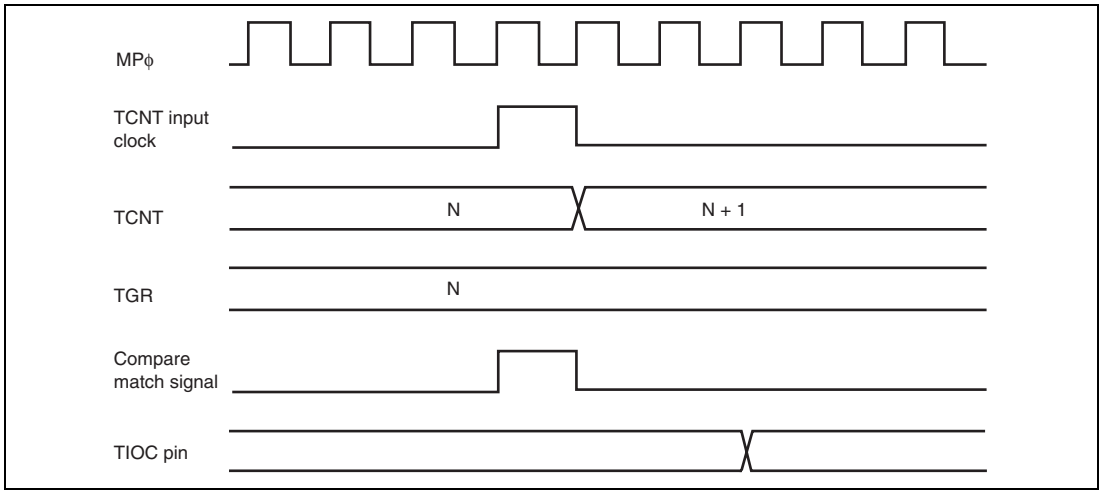



**Figure 10.98 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)**

**Input Capture Signal Timing:** Figure 10.99 shows input capture signal timing.

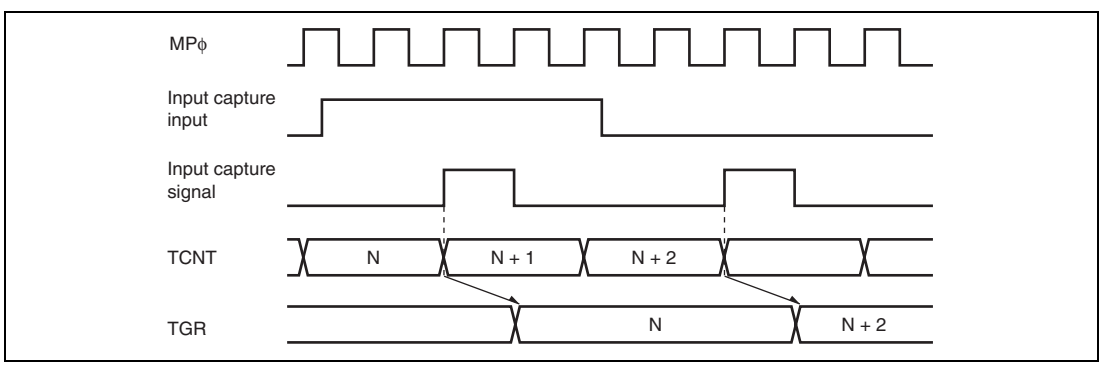



**Figure 10.99 Input Capture Input Signal Timing**

**Timing for Counter Clearing by Compare Match/Input Capture:** Figures 10.100 and 10.101 show the timing when counter clearing on compare match is specified, and figure 10.102 shows the timing when counter clearing on input capture is specified.

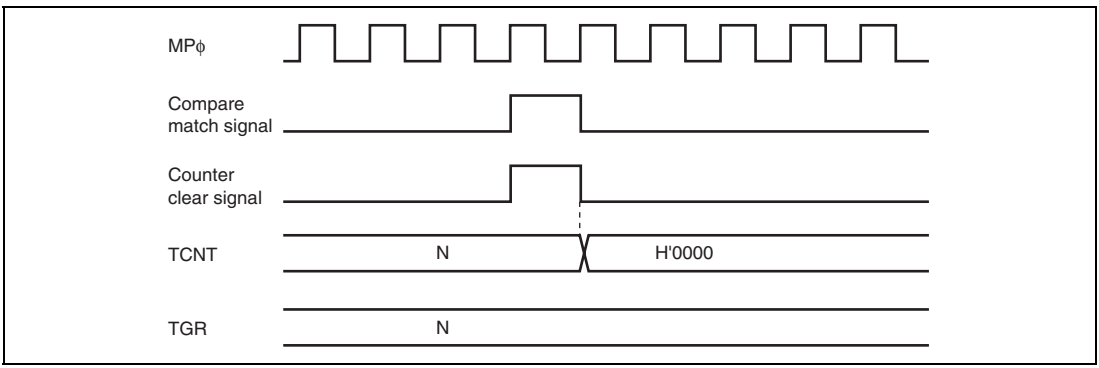



**Figure 10.100 Counter Clear Timing (Compare Match) (Channels 0 to 4)**

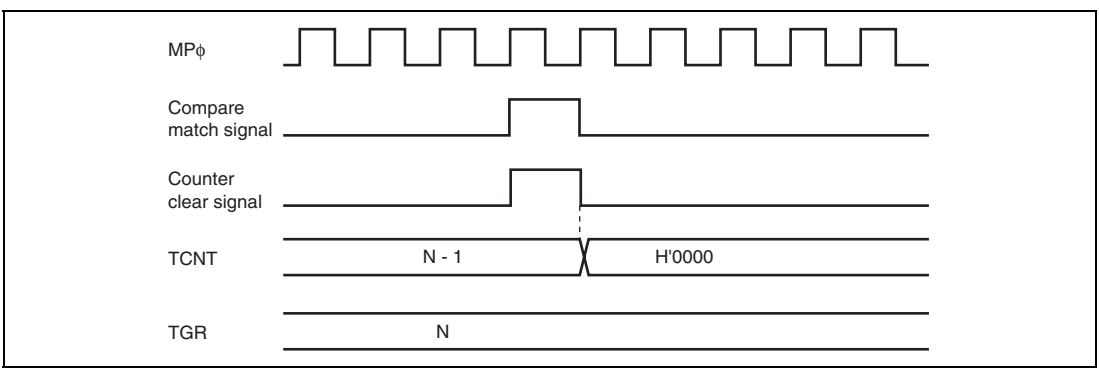



**Figure 10.101 Counter Clear Timing (Compare Match) (Channel 5)**

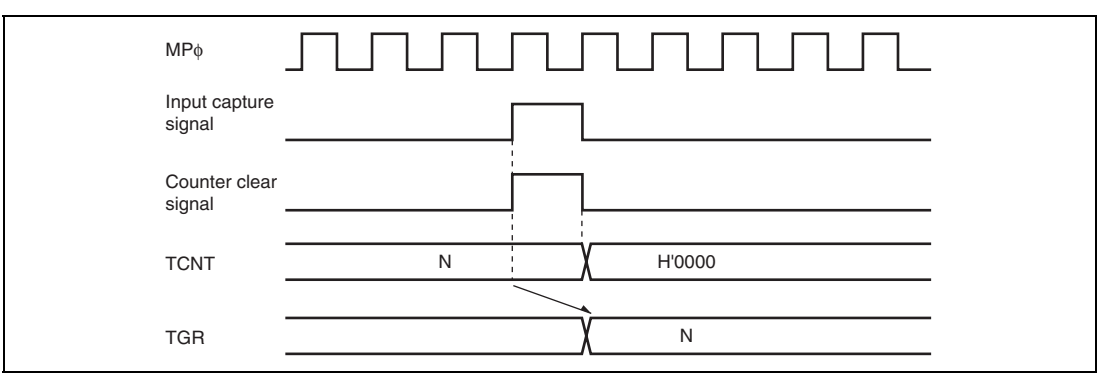



**Figure 10.102 Counter Clear Timing (Input Capture) (Channels 0 to 5)**

**Buffer Operation Timing:** Figures 10.103 to 10.105 show the timing in buffer operation.

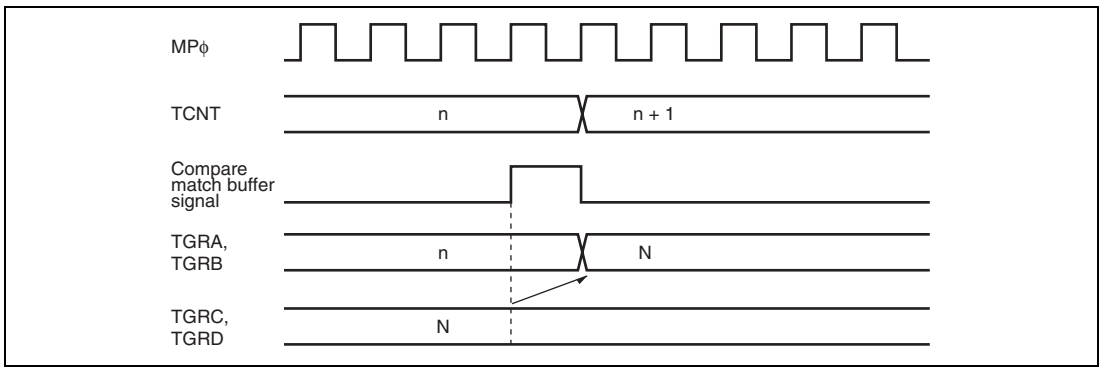



**Figure 10.103 Buffer Operation Timing (Compare Match)**

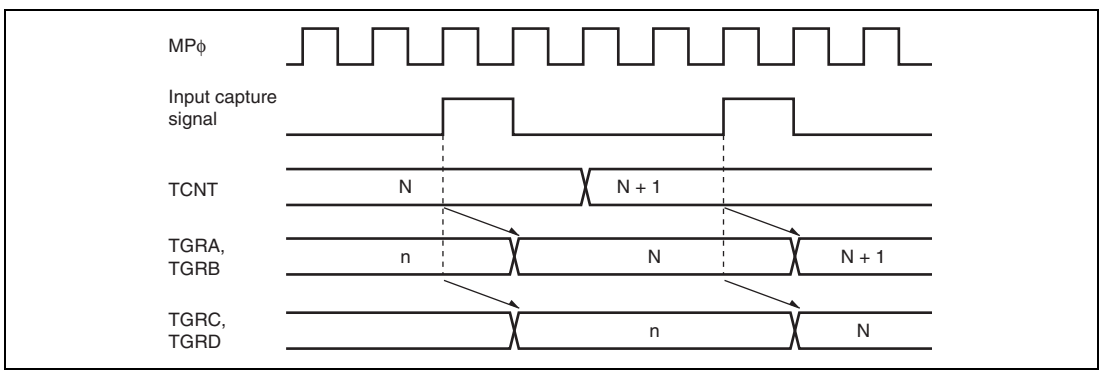



**Figure 10.104 Buffer Operation Timing (Input Capture)**

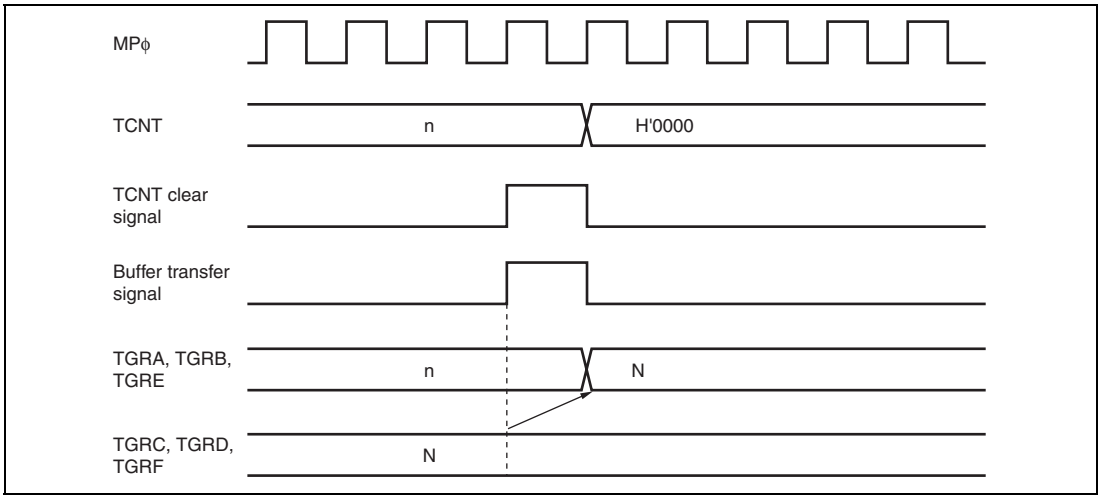



**Figure 10.105 Buffer Transfer Timing (when TCNT Cleared)**

**Buffer Transfer Timing (Complementary PWM Mode)**: Figures 10.106 to 10.108 show the buffer transfer timing in complementary PWM mode.

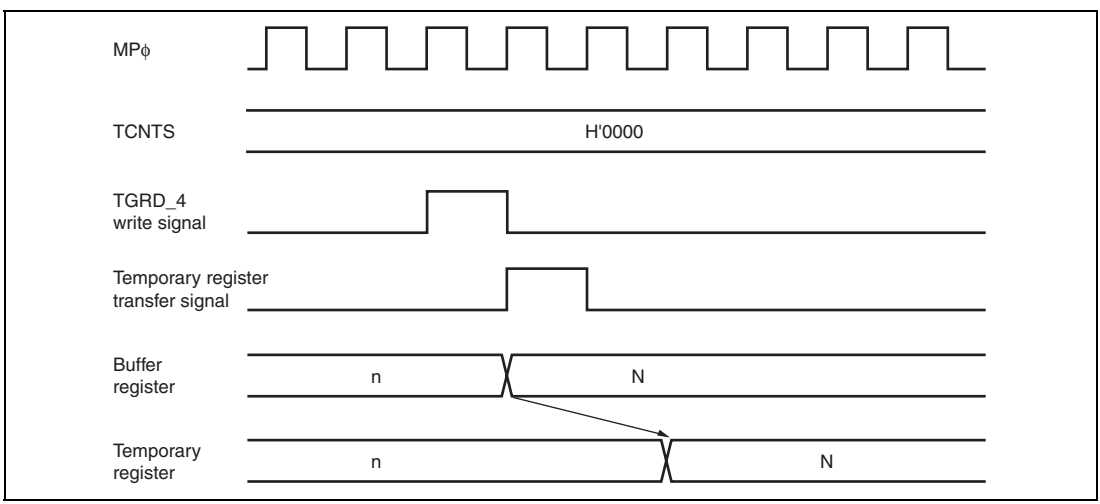



**Figure 10.106 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)**

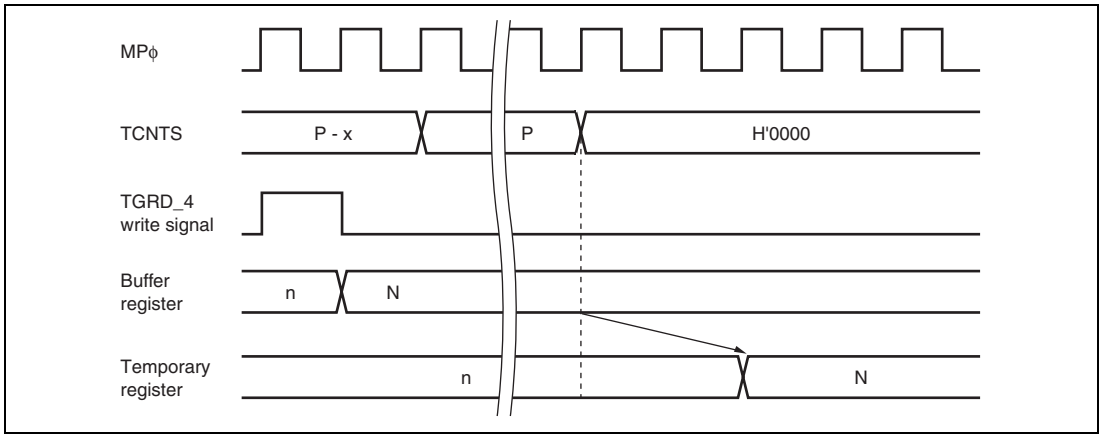



**Figure 10.107 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)**

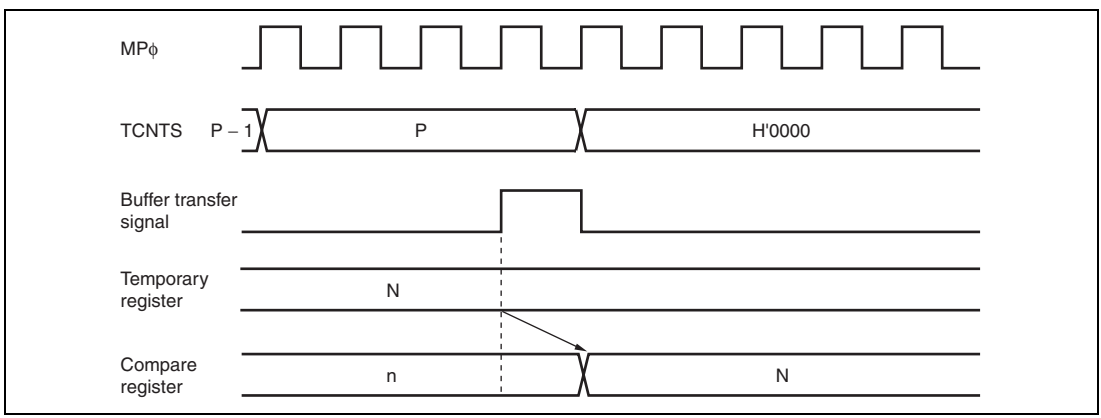



**Figure 10.108 Transfer Timing from Temporary Register to Compare Register**

### 10.6.2 Interrupt Signal Timing

**TGF Flag Setting Timing in Case of Compare Match:** Figures 10.109 and 10.110 show the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

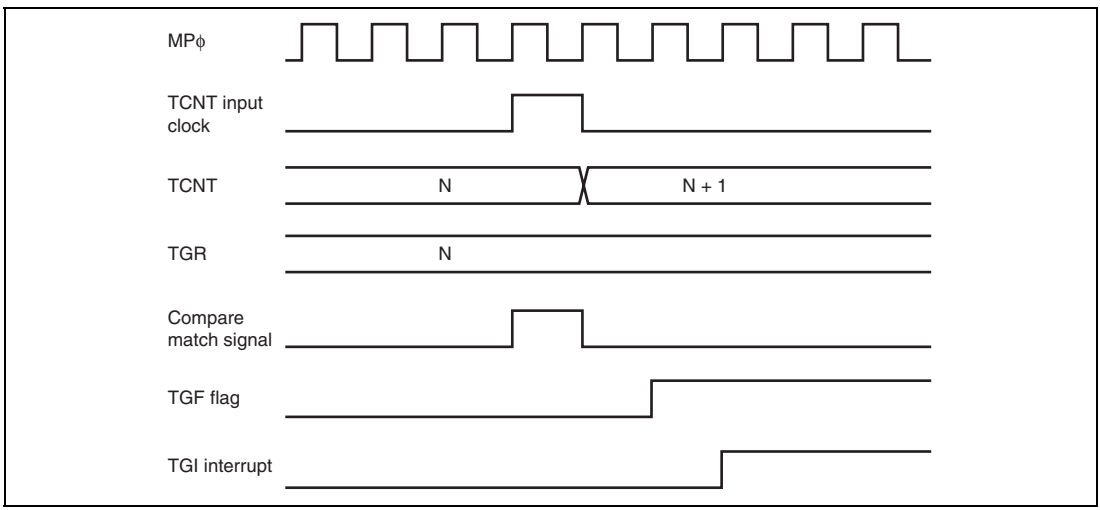



**Figure 10.109 TGI Interrupt Timing (Compare Match) (Channels 0 to 4)**

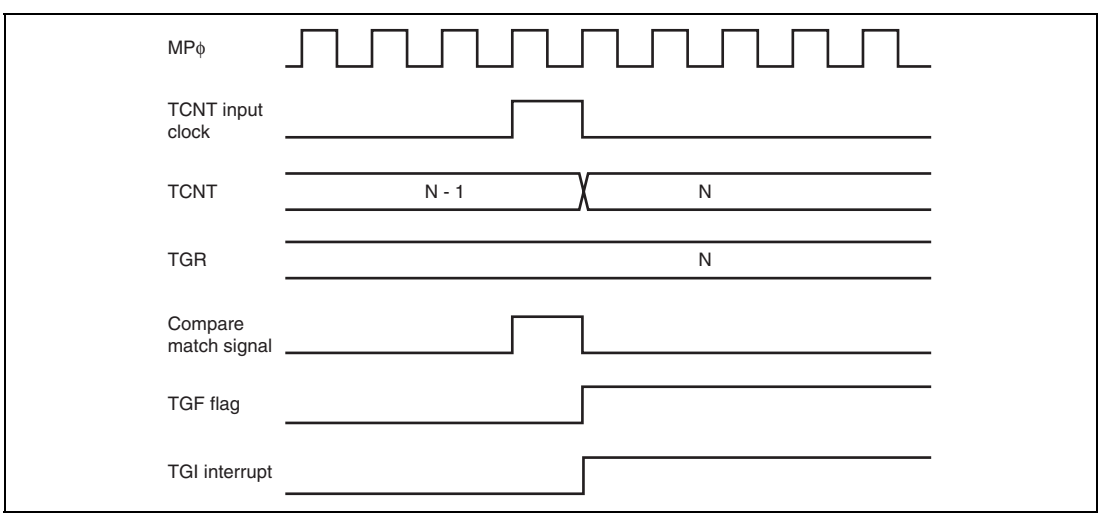



**Figure 10.110 TGI Interrupt Timing (Compare Match) (Channel 5)**

**TGF Flag Setting Timing in Case of Input Capture:** Figures 10.111 and 10.112 show the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

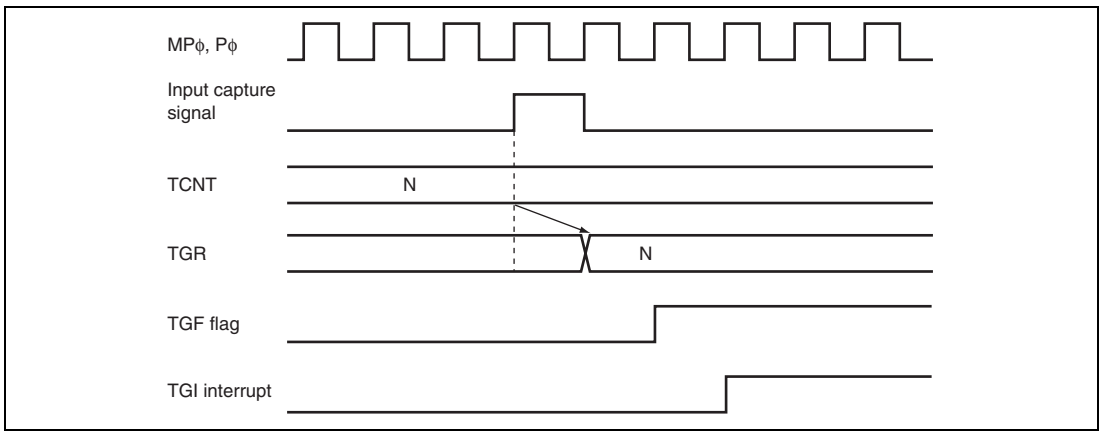



**Figure 10.111 TGI Interrupt Timing (Input Capture) (Channels 0 to 4)**

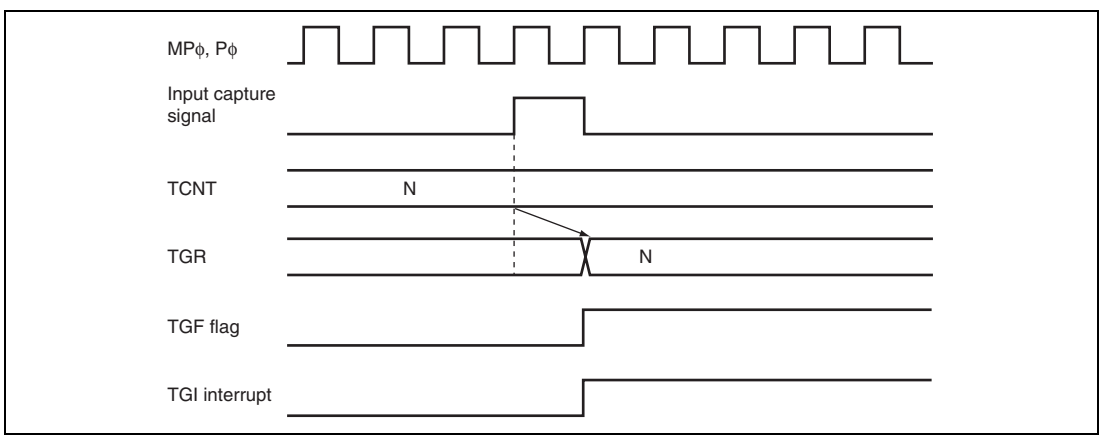



**Figure 10.112 TGI Interrupt Timing (Input Capture) (Channel 5)**

**TCFV Flag/TCFU Flag Setting Timing:** Figure 10.113 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 10.114 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

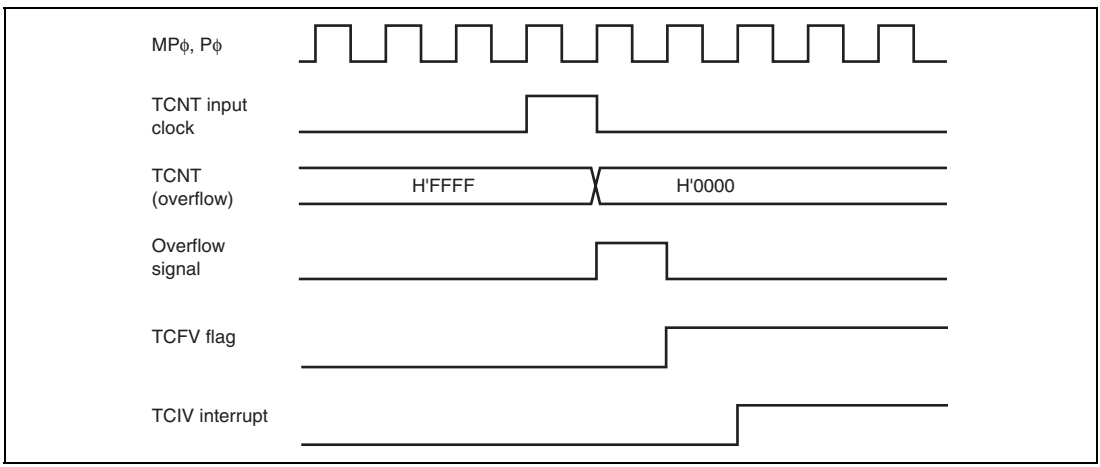



**Figure 10.113 TCIV Interrupt Setting Timing**

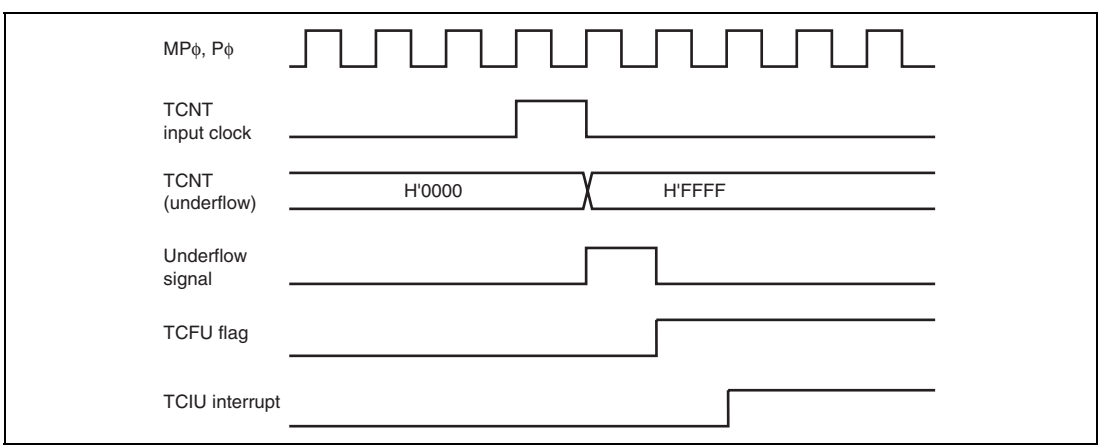



**Figure 10.114 TCIU Interrupt Setting Timing**

**Status Flag Clearing Timing:** After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC is activated, the flag is cleared automatically. Figures 10.115 and 10.116 show the timing for status flag clearing by the CPU, and figures 10.117 and 10.118 show the timing for status flag clearing by the DTC.

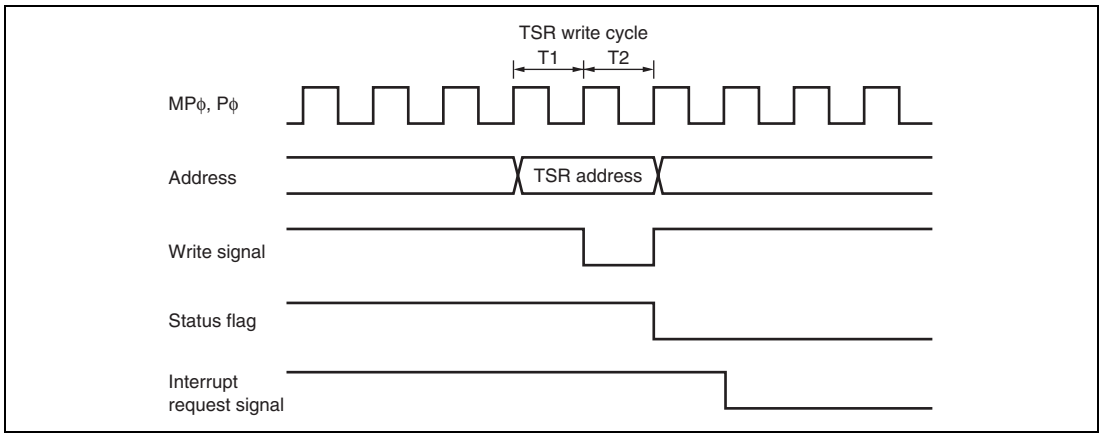



**Figure 10.115 Timing for Status Flag Clearing by CPU (Channels 0 to 4)**

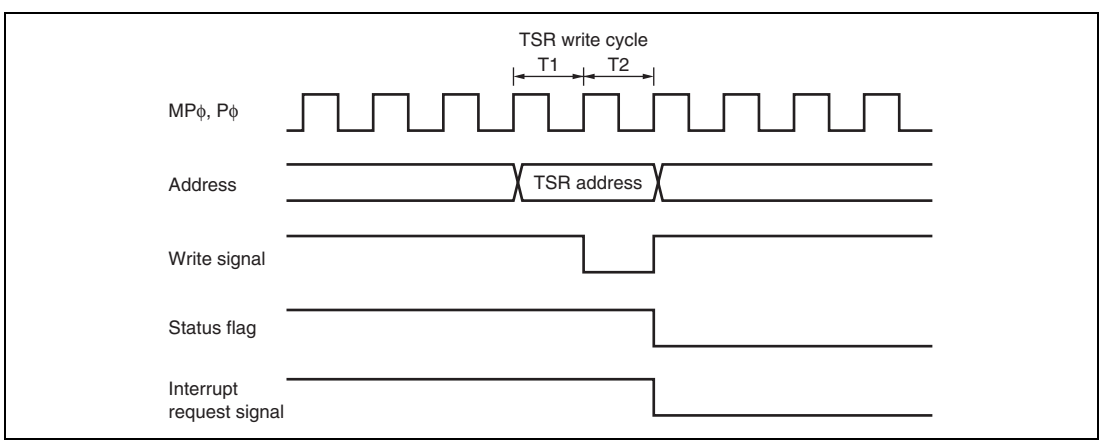



**Figure 10.116 Timing for Status Flag Clearing by CPU (Channel 5)**

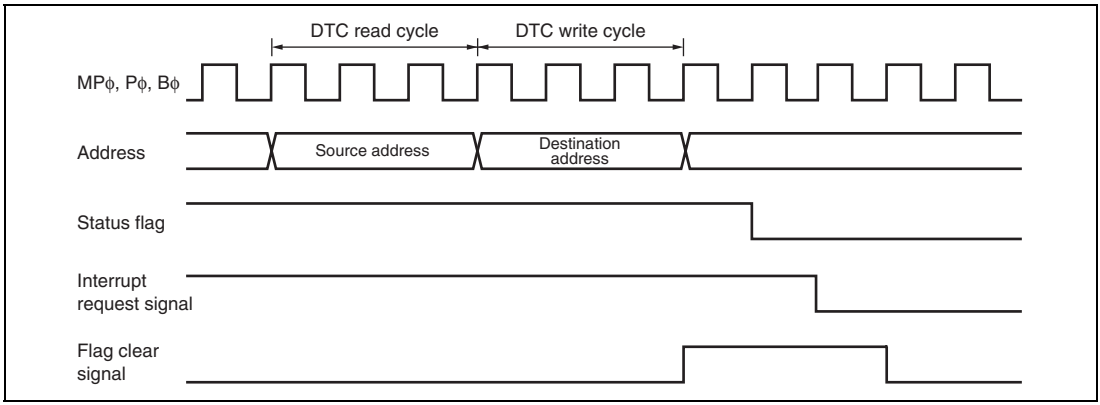



**Figure 10.117 Timing for Status Flag Clearing by DTC Activation (Channels 0 to 4)**

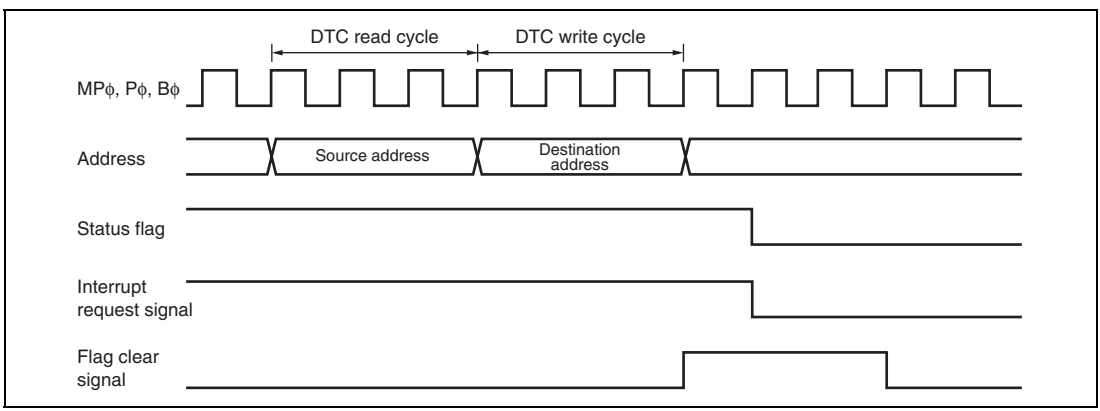



**Figure 10.118 Timing for Status Flag Clearing by DTC Activation (Channel 5)**

## 10.7 Usage Notes

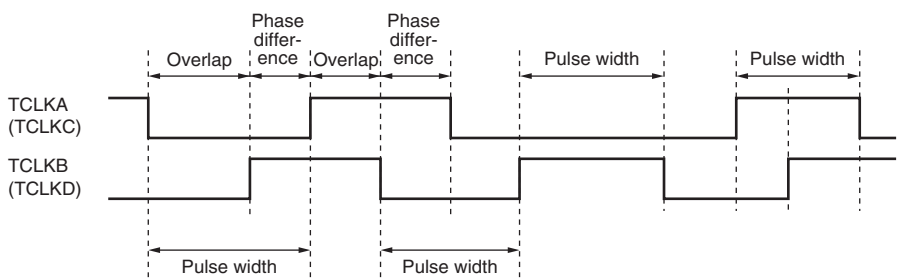
### 10.7.1 Module Standby Mode Setting

MTU2 operation can be disabled or enabled using the standby control register. The initial setting is for MTU2 operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes.

### 10.7.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The MTU2 will not operate properly at narrower pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.119 shows the input clock conditions in phase counting mode.




Notes: Phase difference and overlap : 1.5 states or more
Pulse width : 2.5 states or more

**Figure 10.119 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode**

### 10.7.3 Caution on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

- Channels 0 to 4

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

- Channel 5

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

Where f: Counter frequency
MPϕ: MTU2 peripheral clock operating frequency
N: TGR set value

### 10.7.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 10.120 shows the timing in this case.

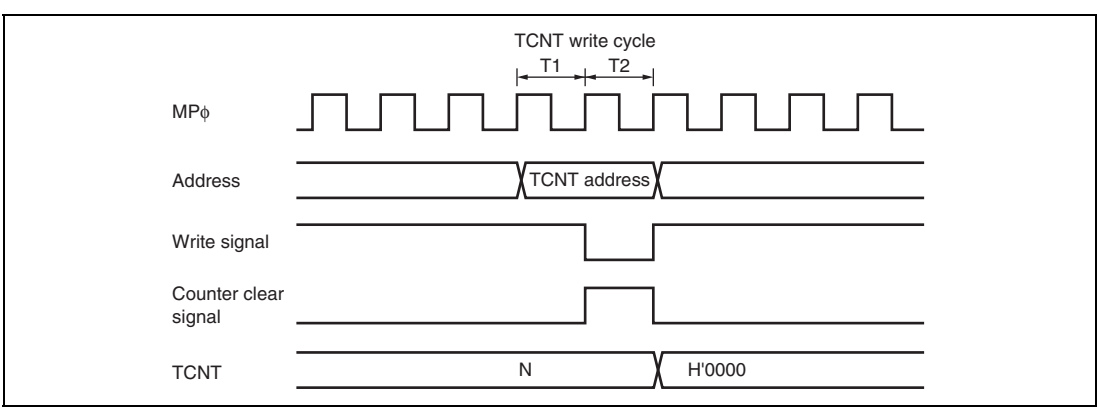



**Figure 10.120 Contention between TCNT Write and Clear Operations**

### 10.7.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 10.121 shows the timing in this case.

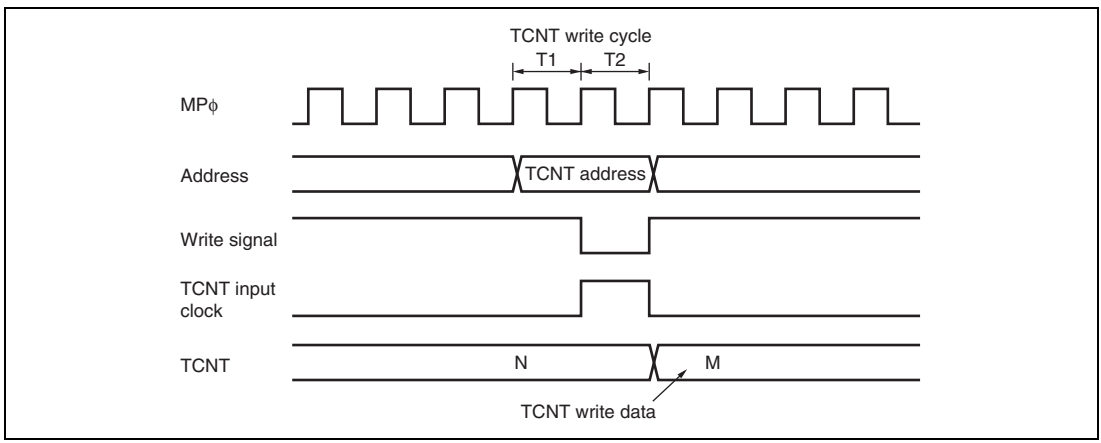



**Figure 10.121 Contention between TCNT Write and Increment Operations**

### 10.7.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write is executed and the compare match signal is also generated.

Figure 10.122 shows the timing in this case.

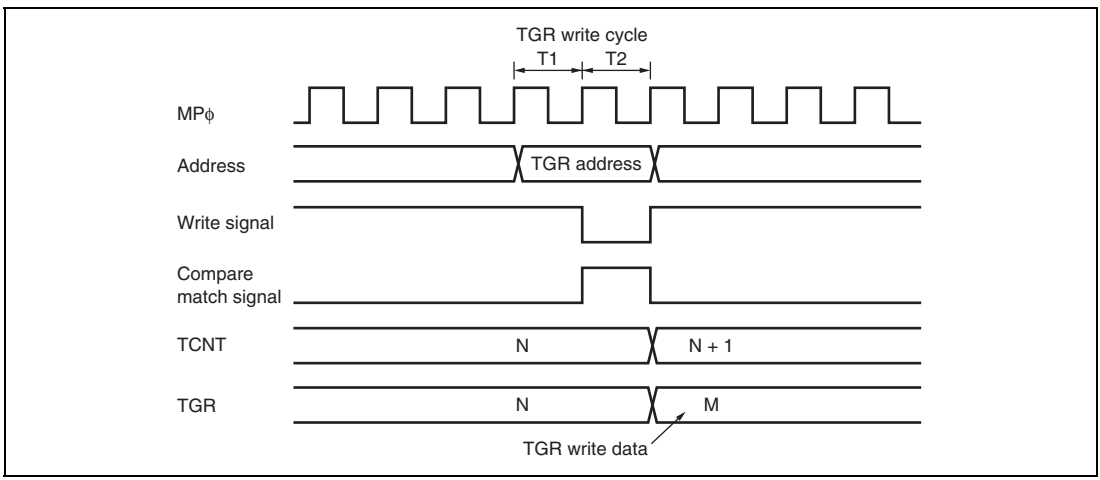



**Figure 10.122 Contention between TGR Write and Compare Match**

### 10.7.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data before write.

Figure 10.123 shows the timing in this case.

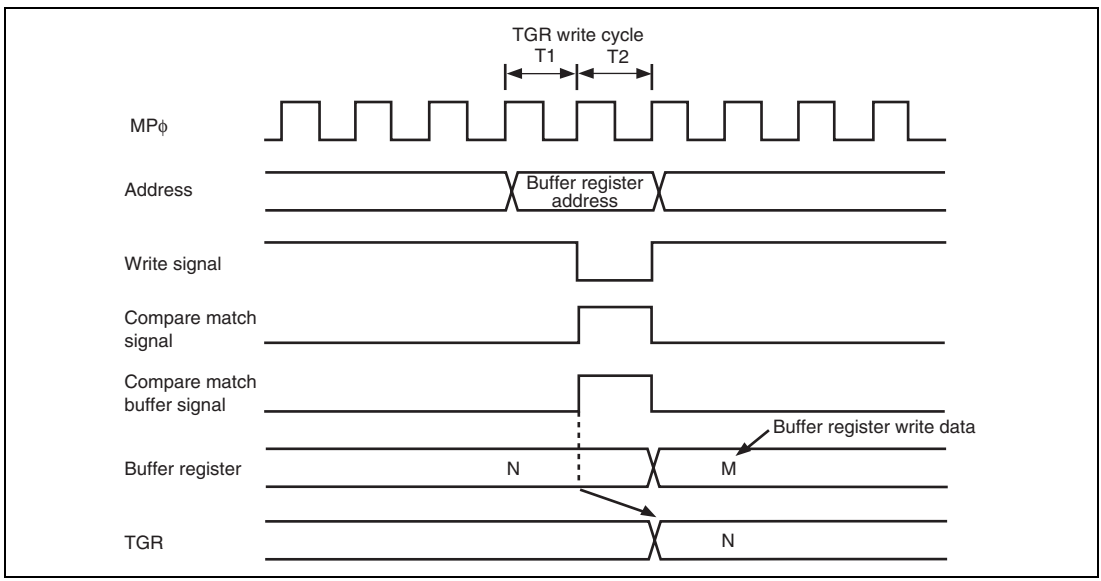



**Figure 10.123 Contention between Buffer Register Write and Compare Match**

### 10.7.8 Contention between Buffer Register Write and TCNT Clear

When the buffer transfer timing is set at the TCNT clear by the buffer transfer mode register (TBTM), if TCNT clear occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data before write.

Figure 10.124 shows the timing in this case.

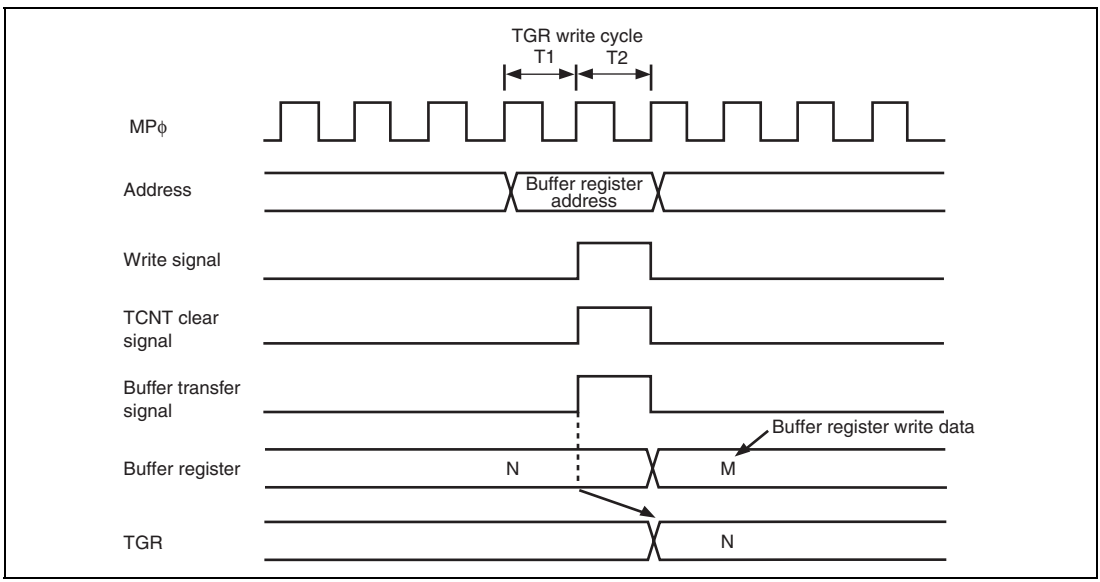



**Figure 10.124 Contention between Buffer Register Write and TCNT Clear**

### 10.7.9 Contention between TGR Read and Input Capture

If an input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data in the buffer before input capture transfer for channels 0 to 4, and the data after input capture transfer for channel 5.

Figures 10.125 and 10.126 show the timing in this case.

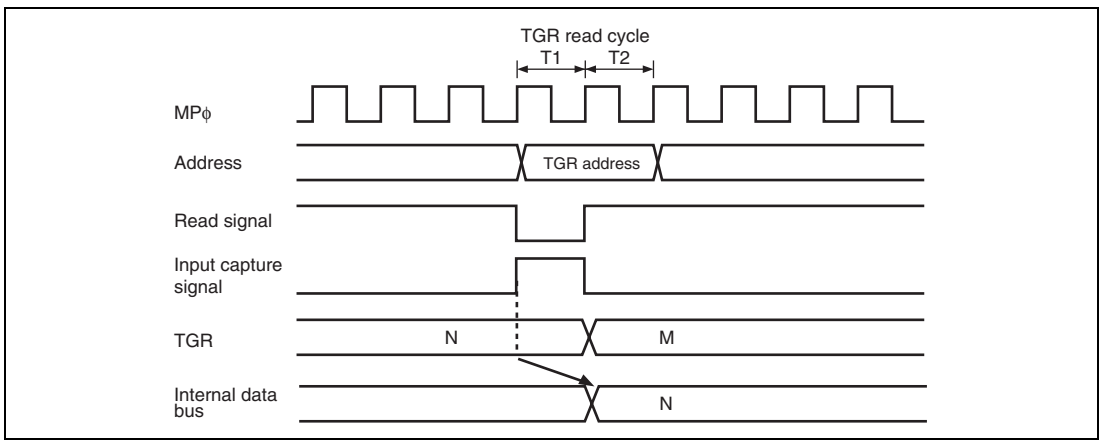



**Figure 10.125 Contention between TGR Read and Input Capture (Channels 0 to 4)**

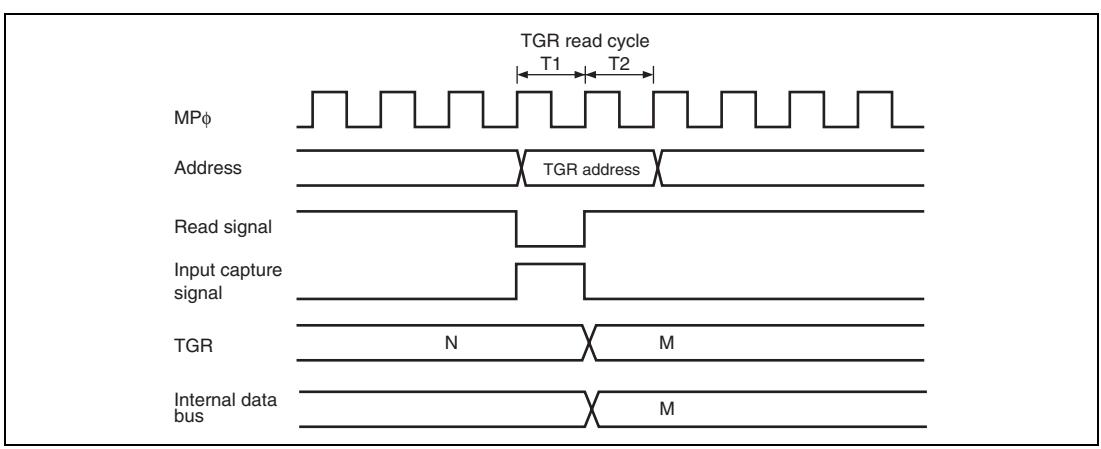



**Figure 10.126 Contention between TGR Read and Input Capture (Channel 5)**

### 10.7.10 Contention between TGR Write and Input Capture

If an input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed for channels 0 to 4. For channel 5, write to TGR is performed and the input capture signal is generated.

Figures 10.127 and 10.128 show the timing in this case.

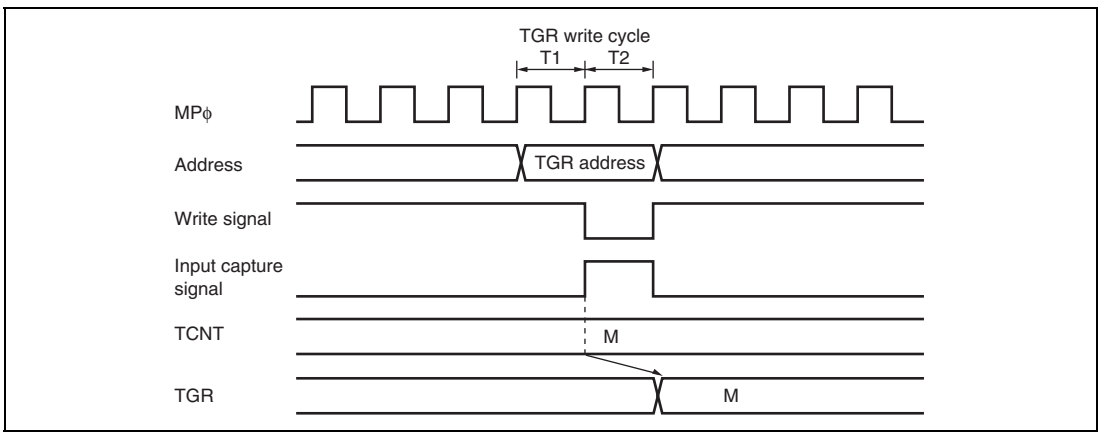



**Figure 10.127 Contention between TGR Write and Input Capture (Channels 0 to 4)**

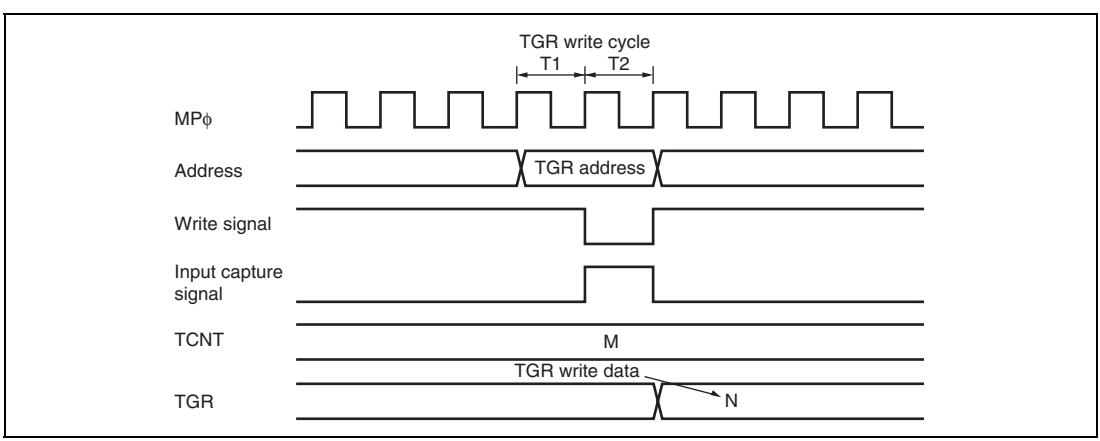



**Figure 10.128 Contention between TGR Write and Input Capture (Channel 5)**

### 10.7.11 Contention between Buffer Register Write and Input Capture

If an input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 10.129 shows the timing in this case.

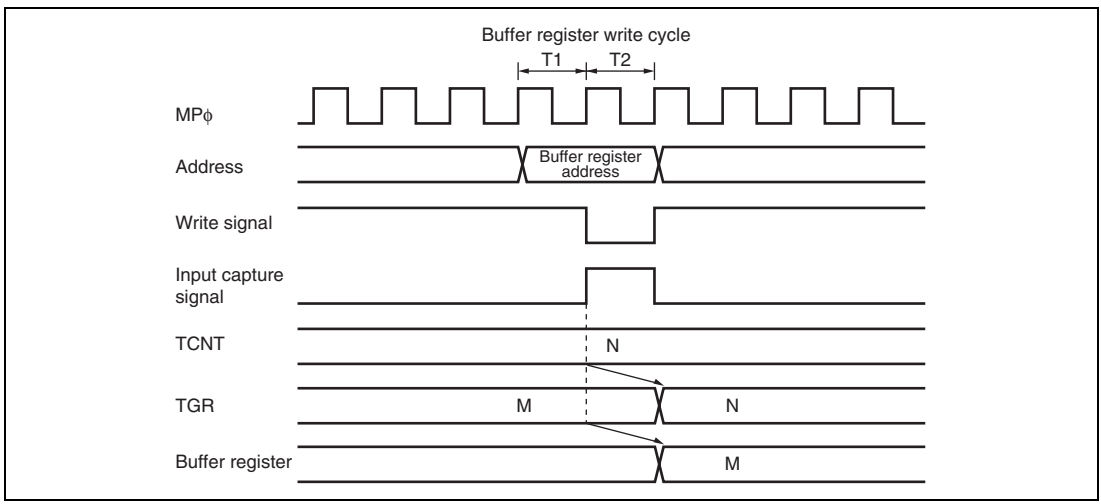



**Figure 10.129 Contention between Buffer Register Write and Input Capture**

### 10.7.12 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 10.130.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

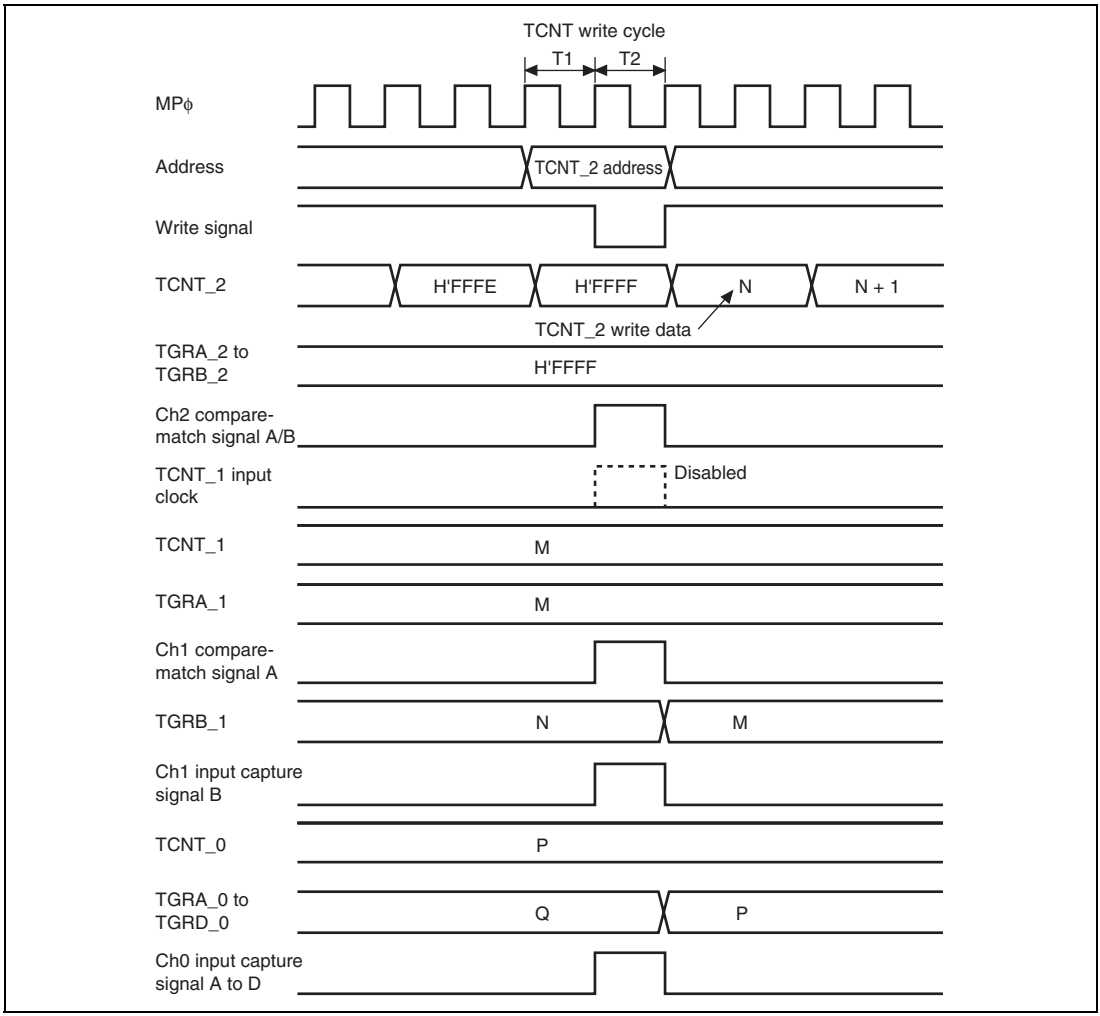



**Figure 10.130 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection**

### 10.7.13 Counter Value during Complementary PWM Mode Stop

When counting operation is suspended with TCNT_3 and TCNT_4 in complementary PWM mode, TCNT_3 has the timer dead time register (TDDR) value, and TCNT_4 is held at H'0000.

When restarting complementary PWM mode, counting begins automatically from the initialized state. This explanatory diagram is shown in figure 10.131.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

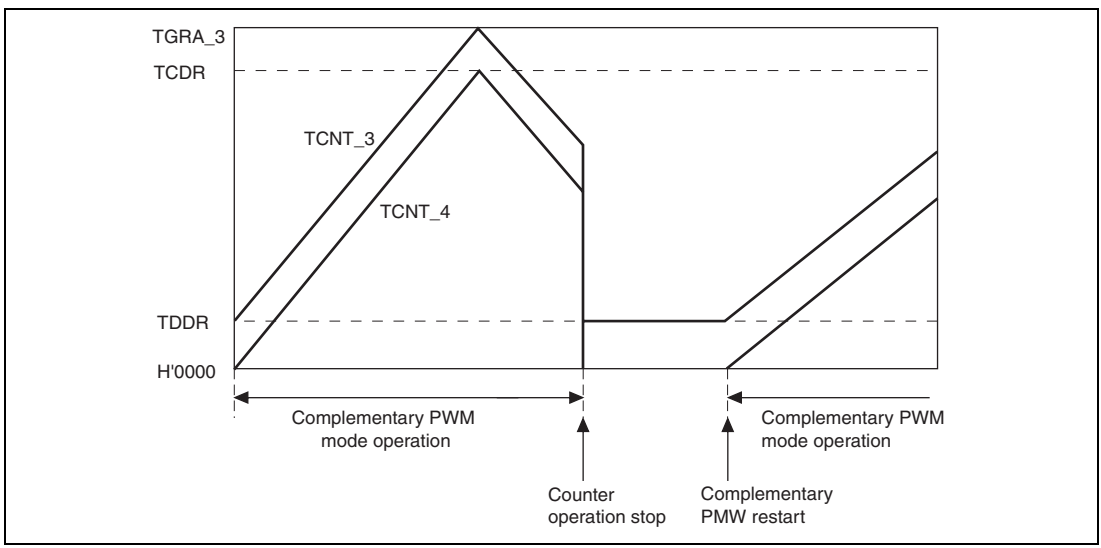



**Figure 10.131 Counter Value during Complementary PWM Mode Stop**

### 10.7.14 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle setting register (TGRA_3), timer cycle data register (TCDR), and duty setting registers (TGRB_3, TGRA_4, and TGRB_4).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance with bit settings BFA and BFB of TMDR_3. When TMDR_3's BFA bit is set to 1, TGRC_3 functions as a buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TGRA_4, and TCBR functions as the TCDR's buffer register.

### 10.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag

When setting buffer operation for reset sync PWM mode, set the BFA and BFB bits 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 10.132 shows an example of operations for TGR_3, TGR_4, TIOC3, and TIOC4, with TMDR_3's BFA and BFB bits set to 1, and TMDR_4's BFA and BFB bits set to 0.

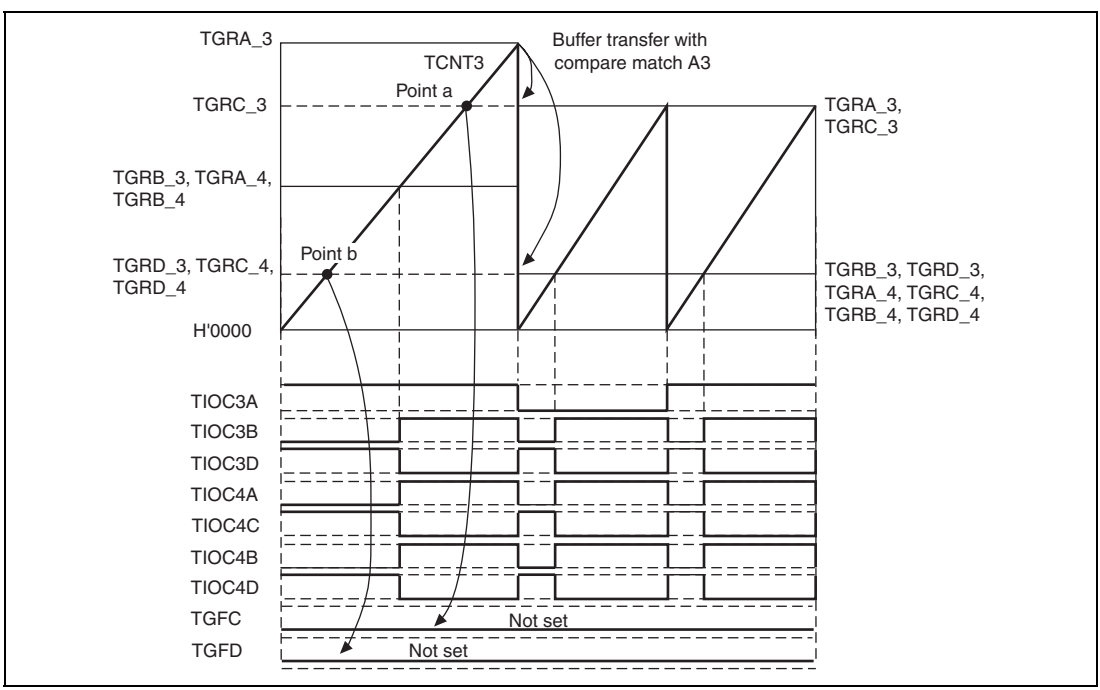



**Figure 10.132 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode**

### 10.7.16 Overflow Flags in Reset Synchronous PWM Mode

When set to reset synchronous PWM mode, TCNT_3 and TCNT_4 start counting when the CST3 bit of TSTR is set to 1. At this point, TCNT_4's count clock source and count edge obey the TCR_3 setting.

In reset synchronous PWM mode, with cycle register TGRA_3's set value at H'FFFF, when specifying TGR3A compare-match for the counter clear source, TCNT_3 and TCNT_4 count up to H'FFFF, then a compare-match occurs with TGRA_3, and TCNT_3 and TCNT_4 are both cleared. At this point, TSR's overflow flag TCFV bit is not set.

Figure 10.133 shows a TCFV bit operation example in reset synchronous PWM mode with a set value for cycle register TGRA_3 of H'FFFF, when a TGRA_3 compare-match has been specified without synchronous setting for the counter clear source.

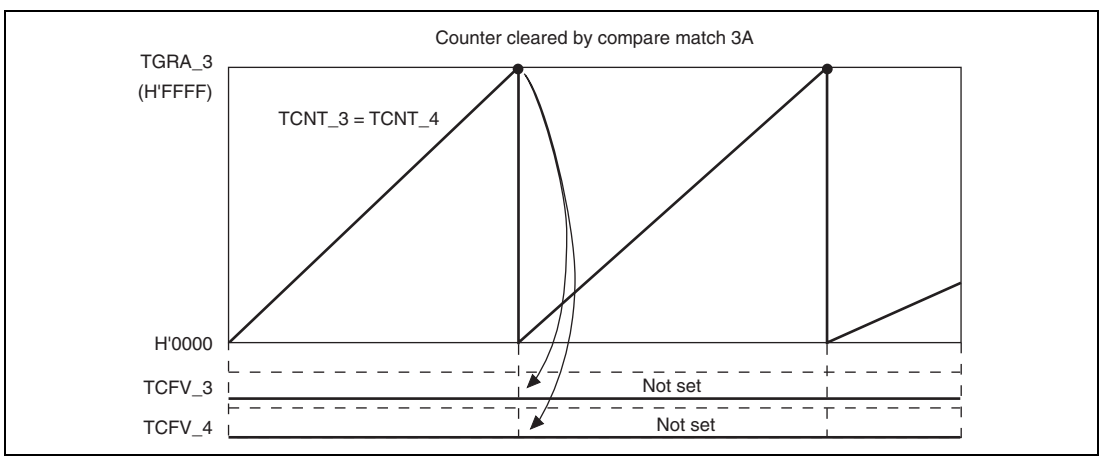



**Figure 10.133 Reset Synchronous PWM Mode Overflow Flag**

### 10.7.17 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 10.134 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

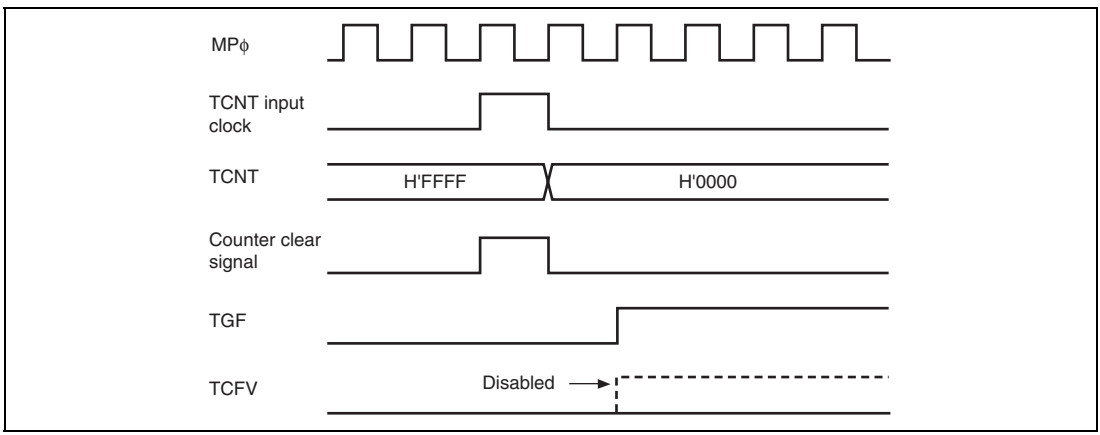



**Figure 10.134 Contention between Overflow and Counter Clearing**

### 10.7.18 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.135 shows the operation timing when there is contention between TCNT write and overflow.

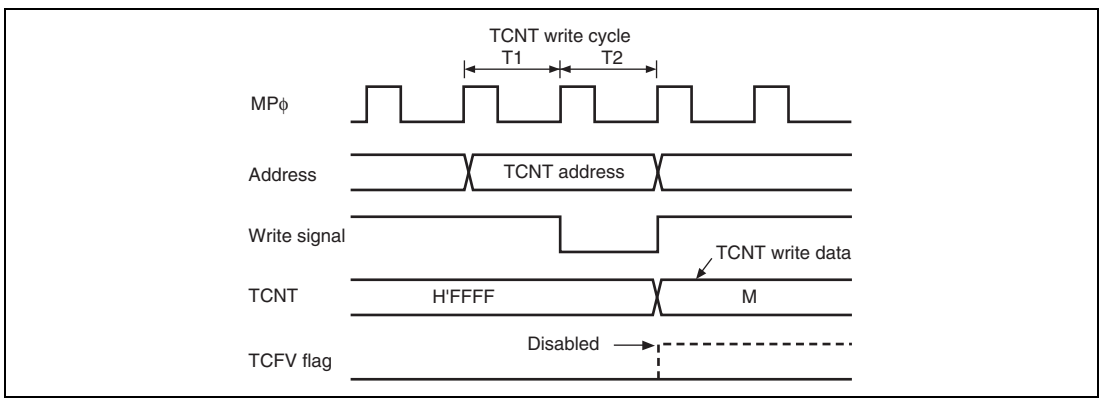



**Figure 10.135 Contention between TCNT Write and Overflow**

### 10.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to reset-synchronized PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-level state, followed by the transition to reset-synchronized PWM mode and operation in that mode, the initial pin output will not be correct.

When making a transition from normal operation to reset-synchronized PWM mode, write H'11 to registers TIORH_3, TIORL_3, TIORH_4, and TIORL_4 to initialize the output pins to low level output, then set an initial register value of H'00 before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, first switch to normal operation, then initialize the output pins to low level output and set an initial register value of H'00 before making the transition to reset-synchronized PWM mode.

### 10.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When channels 3 and 4 are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is set with the OLSP and OLSN bits in the timer output control register (TOCR). In the case of complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to H'00.

### 10.7.21 Interrupts in Module Standby Mode

If module standby mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering module standby mode.

### 10.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection

When timer counters 1 and 2 (TCNT_1 and TCNT_2) are operated as a 32-bit counter in cascade connection, the cascade counter value cannot be captured successfully even if input-capture input is simultaneously done to TIOC1A and TIOC2A or to TIOC1B and TIOC2B. This is because the input timing of TIOC1A and TIOC2A or of TIOC1B and TIOC2B may not be the same when external input-capture signals to be input into TCNT_1 and TCNT_2 are taken in synchronization with the internal clock. For example, TCNT_1 (the counter for upper 16 bits) does not capture the count-up value by overflow from TCNT_2 (the counter for lower 16 bits) but captures the count value before the count-up. In this case, the values of TCNT_1 = H'FFF1 and TCNT_2 = H'0000 should be transferred to TGRA_1 and TGRA_2 or to TGRB_1 and TGRB_2, but the values of TCNT_1 = H'FFF0 and TCNT_2 = H'0000 are erroneously transferred.

The MTU2 has a function that allows simultaneous capture of TCNT_1 and TCNT_2 with a single input-capture input as the trigger. This function allows reading of the 32-bit counter such that TCNT_1 and TCNT_2 are captured at the same time. For details, see section, 10.3.8, Timer Input Capture Control Register (TICCR).

### 10.7.23 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 10.136).

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 10.137).

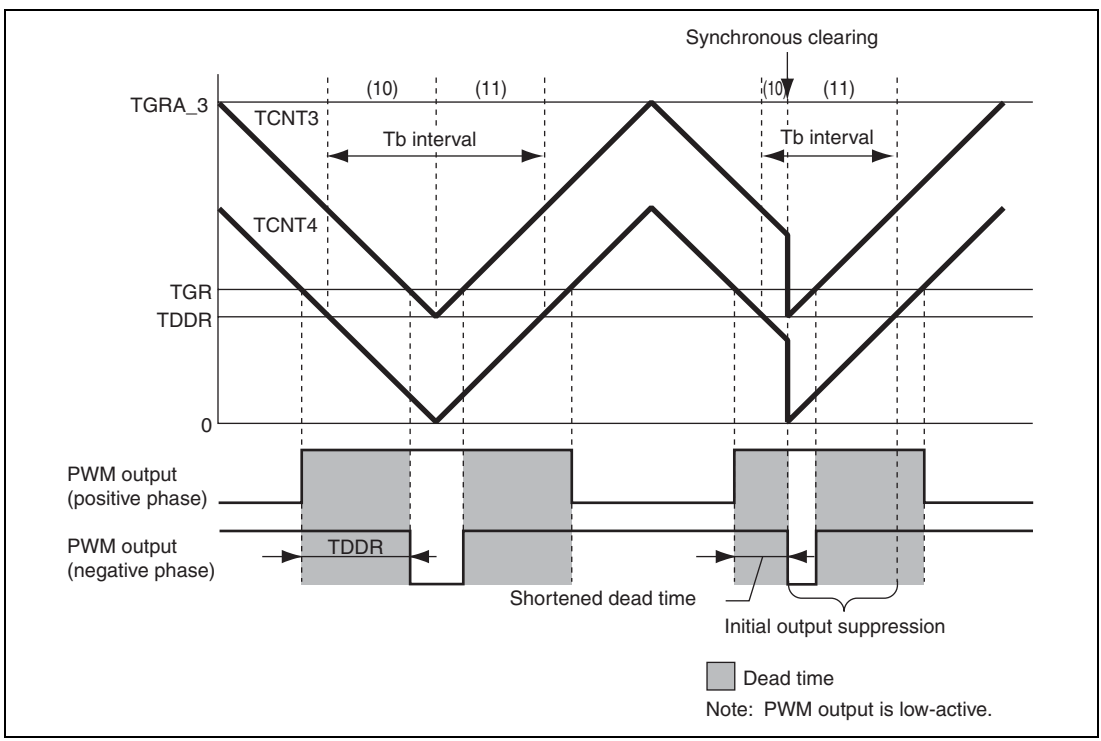



**Figure 10.136 Condition (1) Synchronous Clearing Example**

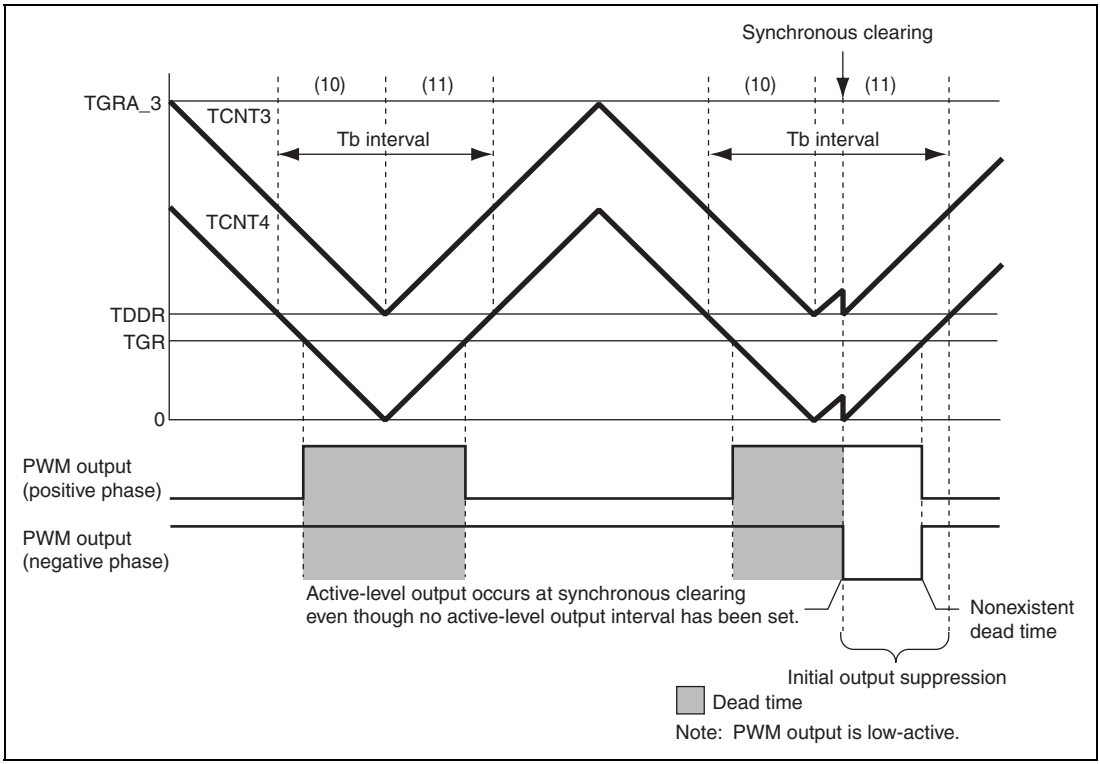



**Figure 10.137 Condition (2) Synchronous Clearing Example**

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.

## 10.8 MTU2 Output Pin Initialization

### 10.8.1 Operating Modes

The MTU2 has the following six operating modes. Waveform output is possible in all of these modes.

- Normal mode (channels 0 to 4)
- PWM mode 1 (channels 0 to 4)
- PWM mode 2 (channels 0 to 2)
- Phase counting modes 1 to 4 (channels 1 and 2)
- Complementary PWM mode (channels 3 and 4)
- Reset-synchronized PWM mode (channels 3 and 4)

The MTU2 output pin initialization method for each of these modes is described in this section.

### 10.8.2 Reset Start Operation

The MTU2 output pins (TIOC*) are initialized low by a reset and in standby mode. Since MTU2 pin function selection is performed by the pin function controller (PFC), when the PFC is set, the MTU2 pin states at that point are output to the ports. When MTU2 output is selected by the PFC immediately after a reset, the MTU2 output initial level, low, is output directly at the port. When the active level is low, the system will operate at this point, and therefore the PFC setting should be made after initialization of the MTU2 output pins is completed.

Note: Channel number and port notation are substituted for *.

### 10.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc.

If an error occurs during MTU2 operation, MTU2 output should be cut by the system. Cutoff is performed by switching the pin output to port output with the PFC and outputting the inverse of the active level. For large-current pins, output can also be cut by hardware, using port output enable (POE). The pin initialization procedures for re-setting due to an error during operation, etc., and the procedures for restarting in a different mode after re-setting, are shown below.

The MTU2 has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Possible mode transition combinations are shown in table 10.59.

**Table 10.59 Mode Transition Combinations**

| | After | | | | | |
|---|---|---|---|---|---|---|
| **Before** | **Normal** | **PWM1** | **PWM2** | **PCM** | **CPWM** | **RPWM** |
| Normal | (1) | (2) | (3) | (4) | (5) | (6) |
| PWM1 | (7) | (8) | (9) | (10) | (11) | (12) |
| PWM2 | (13) | (14) | (15) | (16) | None | None |
| PCM | (17) | (18) | (19) | (20) | None | None |
| CPWM | (21) | (22) | None | None | (23) (24) | (25) |
| RPWM | (26) | (27) | None | None | (28) | (29) |

[Legend]
Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 4
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

### 10.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc.

- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC*B (TIOC *D) pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.
- In PWM mode 2, since a waveform is not output to the cycle register pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode and perform initialization with TIOR, then restore TIOR to its initial value, and temporarily disable channel 3 and 4 output with the timer output master enable register (TOER). Then operate the unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, TOER setting).

Note: Channel number is substituted for * indicated in this article.

Pin initialization procedures are described below for the numbered combinations in table 10.59. The active level is assumed to be low.

**Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode:** Figure 10.138 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

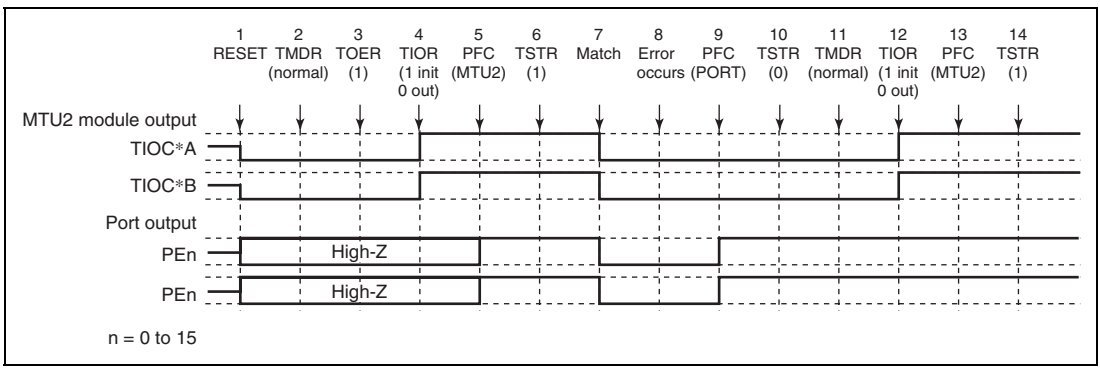



**Figure 10.138 Error Occurrence in Normal Mode, Recovery in Normal Mode**

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Not necessary when restarting in normal mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

**Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 1:** Figure 10.139 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

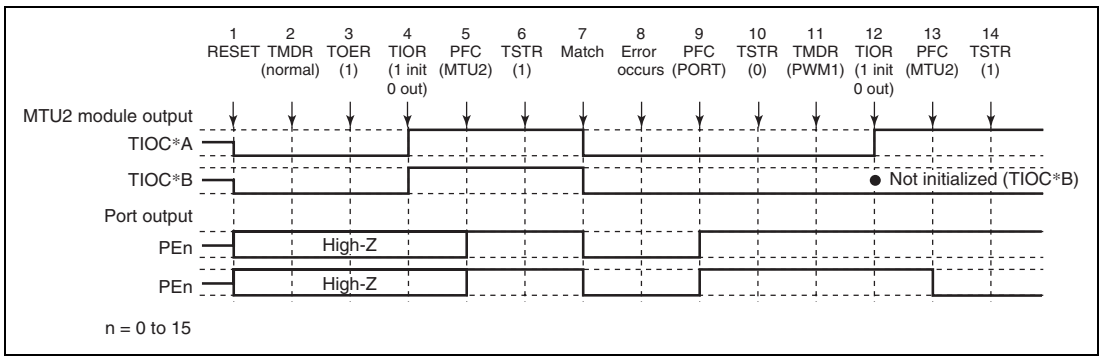



**Figure 10.139 Error Occurrence in Normal Mode, Recovery in PWM Mode 1**

1 to 10 are the same as in figure 10.138.

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.

**Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 2:** Figure 10.140 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

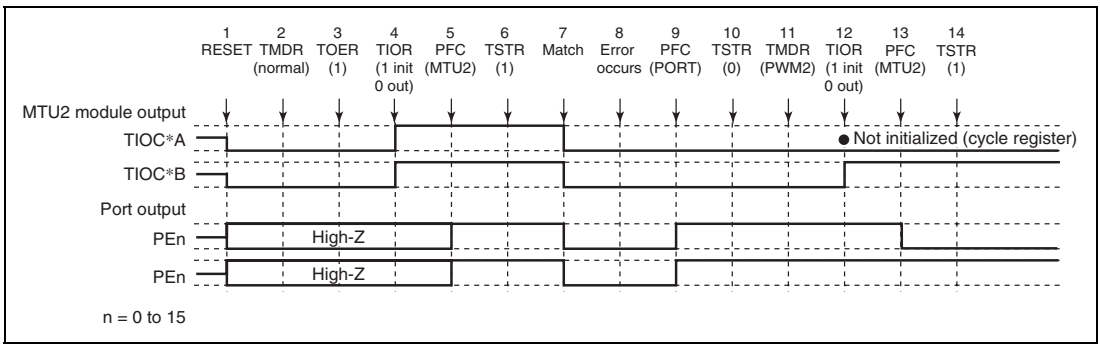



**Figure 10.140 Error Occurrence in Normal Mode, Recovery in PWM Mode 2**

1 to 10 are the same as in figure 10.138.

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.

**Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Phase Counting Mode:** Figure 10.141 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

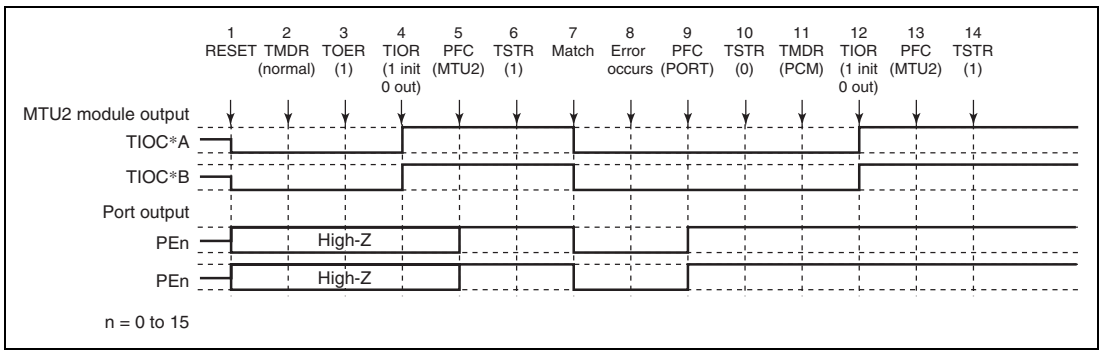



**Figure 10.141 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode**

1 to 10 are the same as in figure 10.138.

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.

**Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Complementary PWM Mode:** Figure 10.142 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

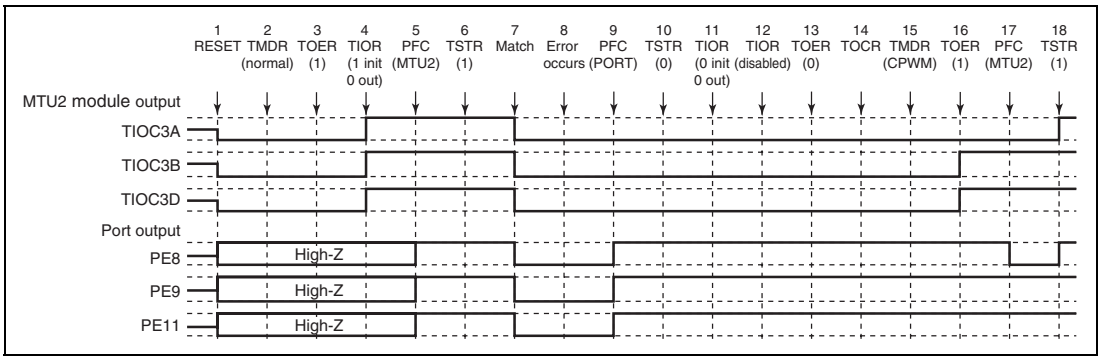



**Figure 10.142 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode**

1 to 10 are the same as in figure 10.138.

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.

**Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode:** Figure 10.143 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

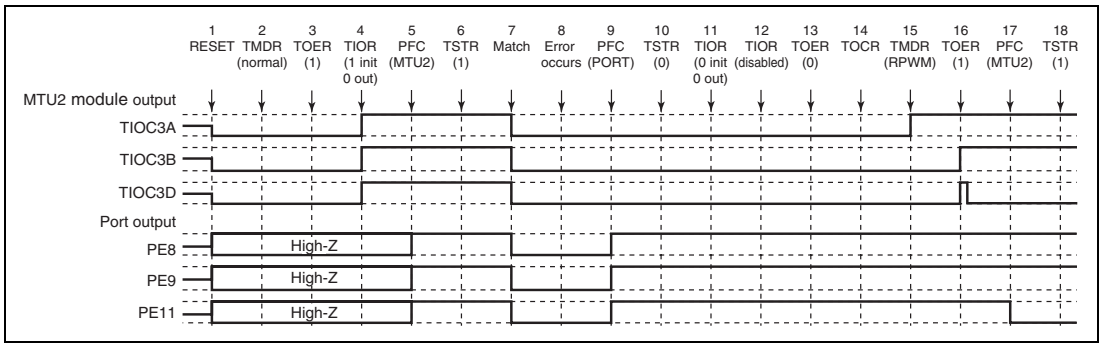



**Figure 10.143 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode**

1 to 13 are the same as in figure 10.138.

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.

**Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Normal Mode:** Figure 10.144 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

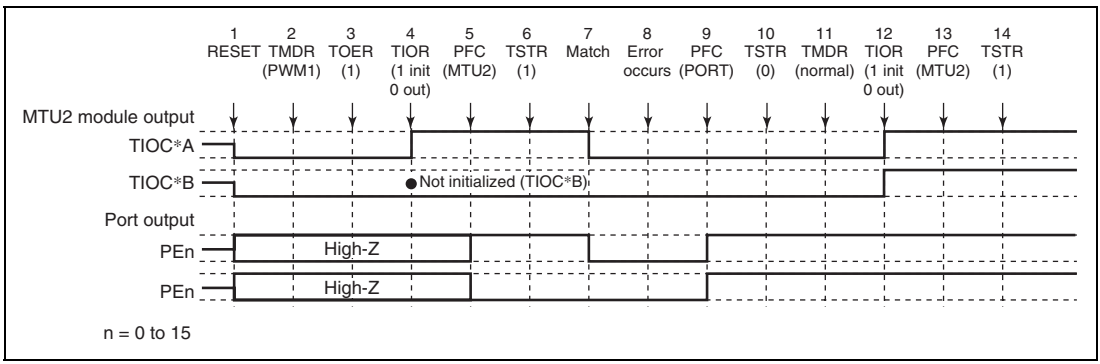



**Figure 10.144 Error Occurrence in PWM Mode 1, Recovery in Normal Mode**

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set PWM mode 1.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 1, the TIOC*B side is not initialized.)
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Set normal mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

**Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 1:** Figure 10.145 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

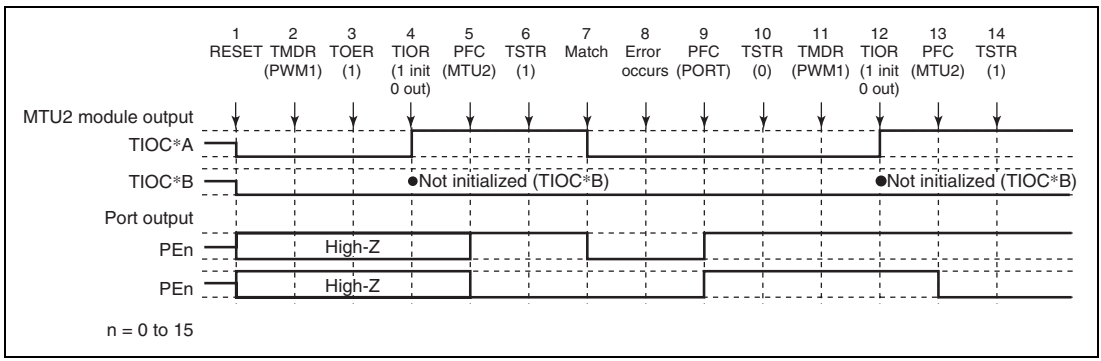



**Figure 10.145 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1**

1 to 10 are the same as in figure 10.144.

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.

**Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 2:** Figure 10.146 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

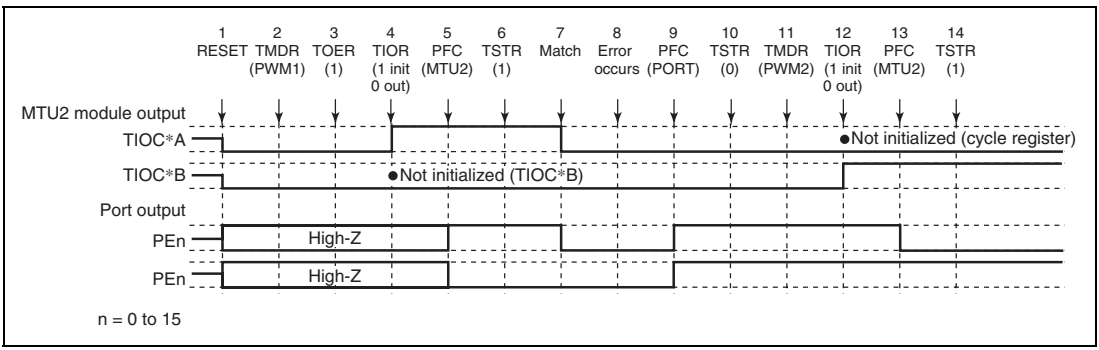



**Figure 10.146 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2**

1 to 10 are the same as in figure 10.144.

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.

**Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Phase Counting Mode:** Figure 10.147 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

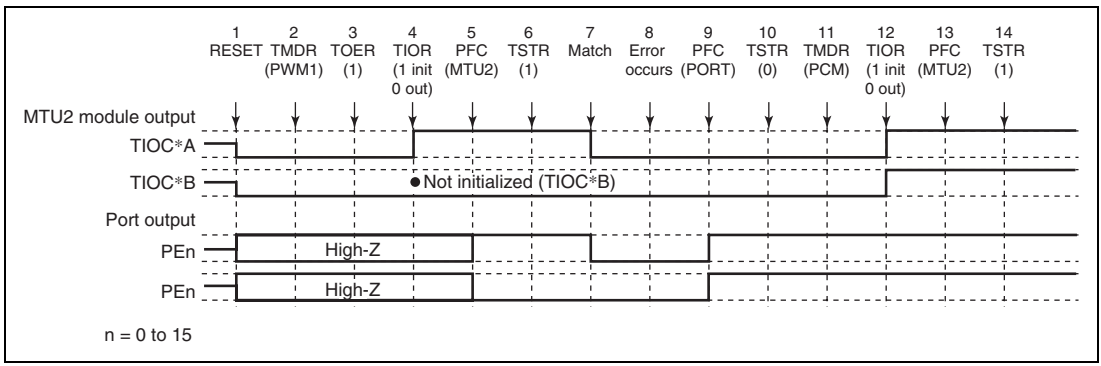



**Figure 10.147 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode**

1 to 10 are the same as in figure 10.144.

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.

**Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Complementary PWM Mode:** Figure 10.148 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

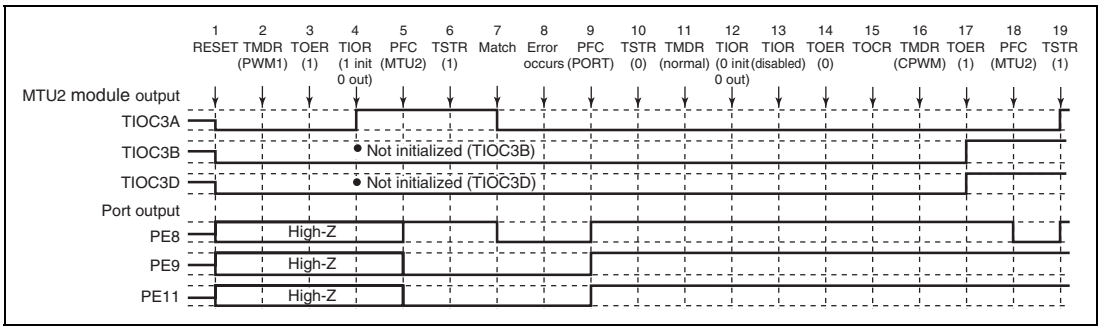



**Figure 10.148 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode**

1 to 10 are the same as in figure 10.144.

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.

**Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Reset-Synchronized PWM Mode:** Figure 10.149 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

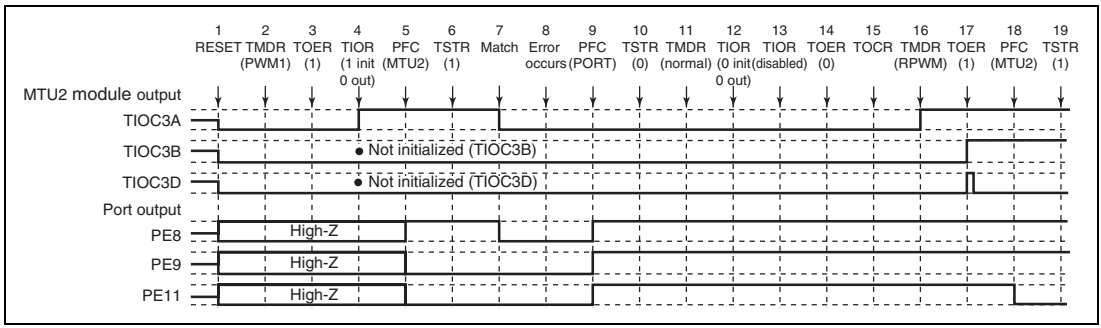



**Figure 10.149 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode**

1 to 14 are the same as in figure 10.148.

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.

**Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Normal Mode:** Figure 10.150 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

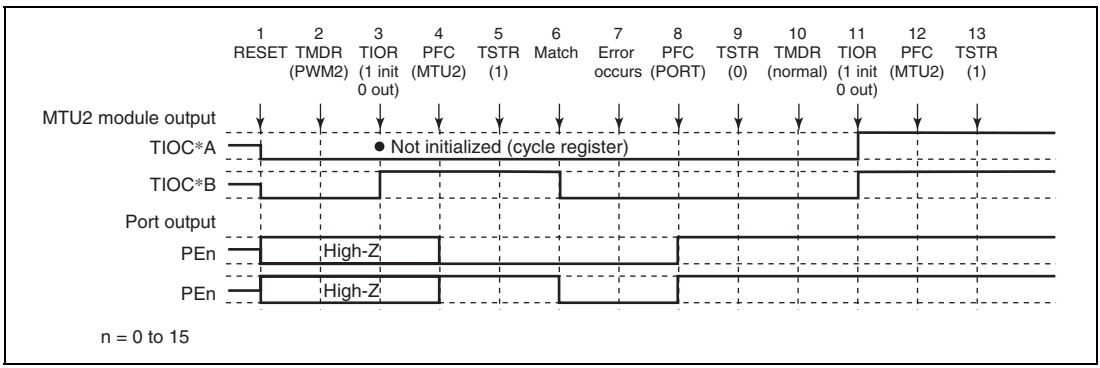



**Figure 10.150 Error Occurrence in PWM Mode 2, Recovery in Normal Mode**

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.

**Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 1:** Figure 10.151 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

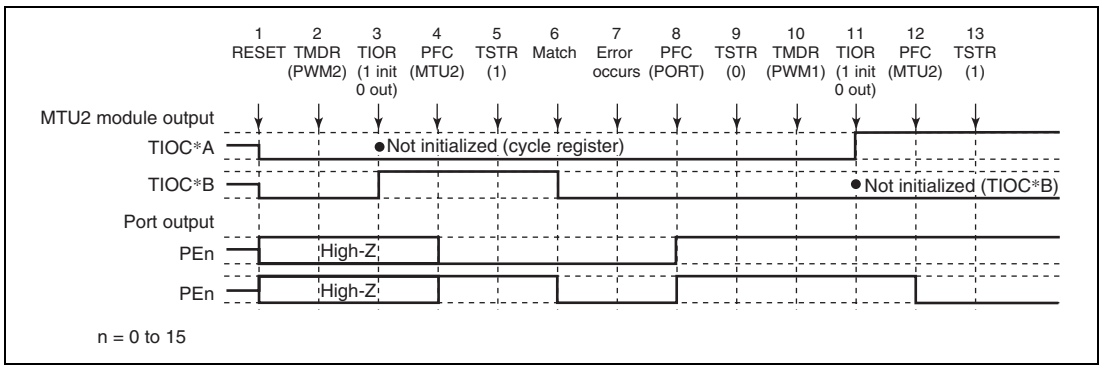



**Figure 10.151 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1**

1 to 9 are the same as in figure 10.150.

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.

**Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 2:** Figure 10.152 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

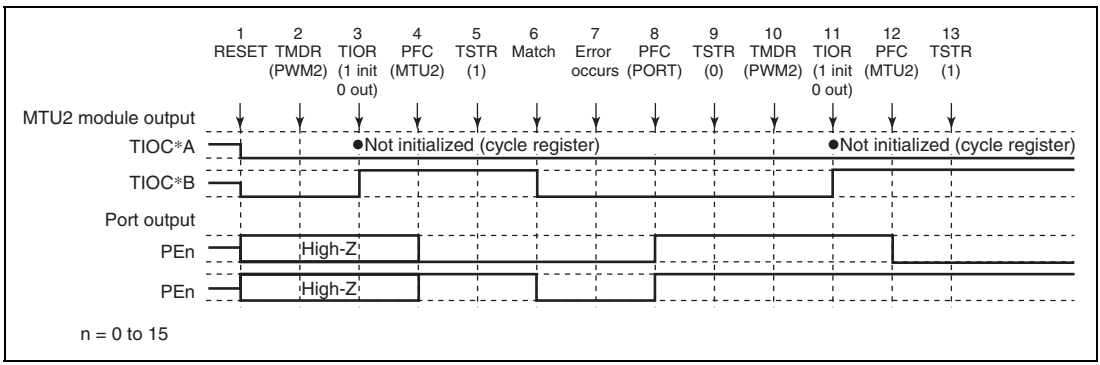



**Figure 10.152 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2**

1 to 9 are the same as in figure 10.150.

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.

**Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Phase Counting Mode:** Figure 10.153 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

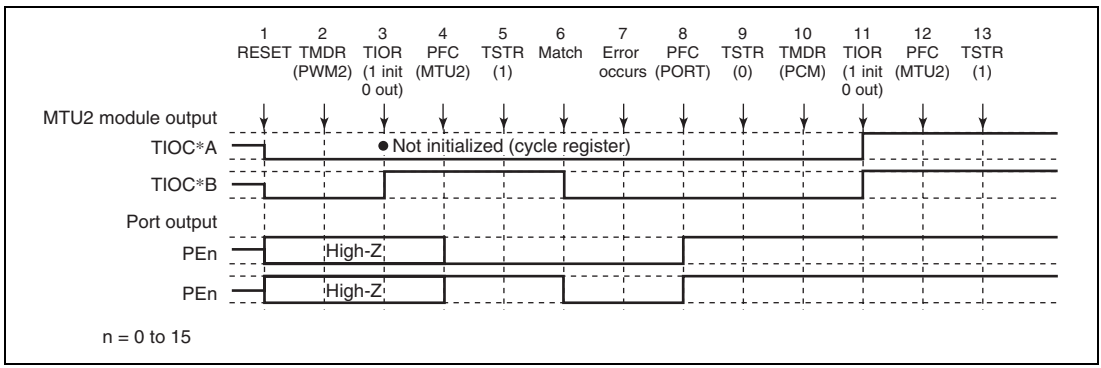



**Figure 10.153 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode**

1 to 9 are the same as in figure 10.150.

10. Set phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

**Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Normal Mode:** Figure 10.154 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

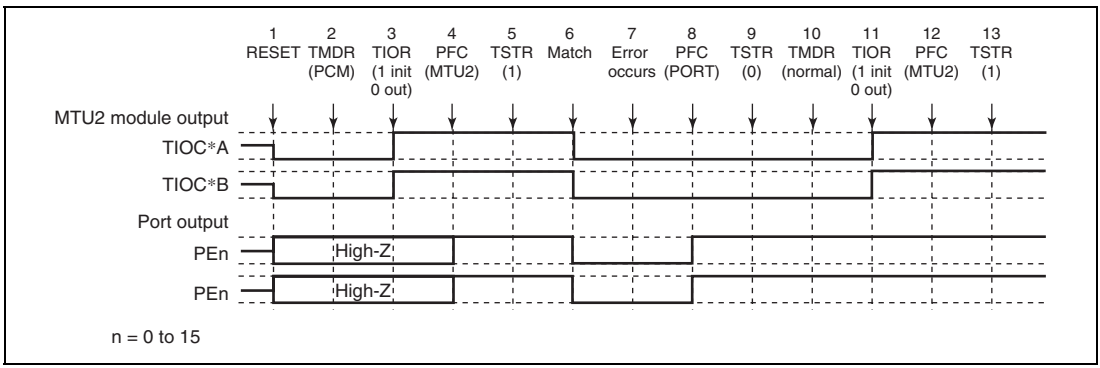



**Figure 10.154 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode**

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set phase counting mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
4. Set MTU2 output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set in normal mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

**Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 1:** Figure 10.155 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

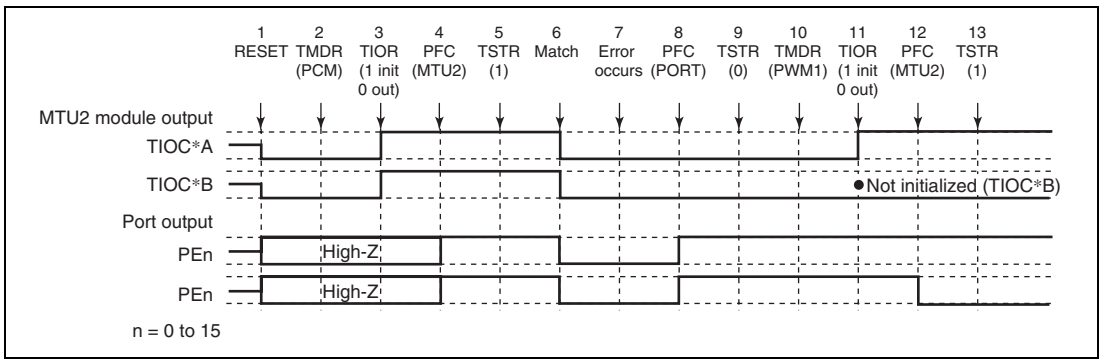



**Figure 10.155 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1**

1 to 9 are the same as in figure 10.154.

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.

**Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 2:** Figure 10.156 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

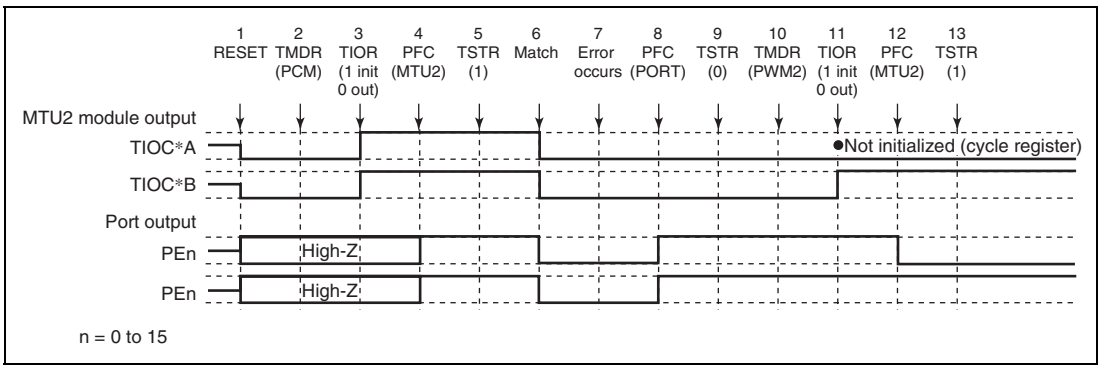



**Figure 10.156 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2**

1 to 9 are the same as in figure 10.154.

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.

**Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Phase Counting Mode:** Figure 10.157 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

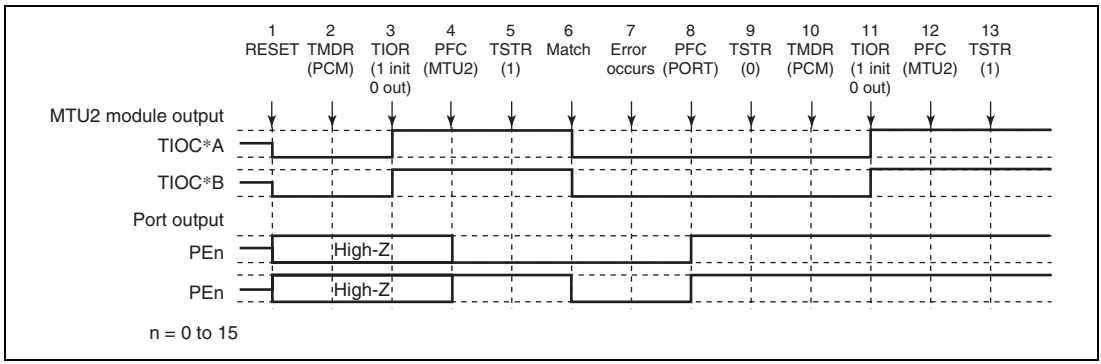



**Figure 10.157 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode**

1 to 9 are the same as in figure 10.154.

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.

**Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Normal Mode:** Figure 10.158 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

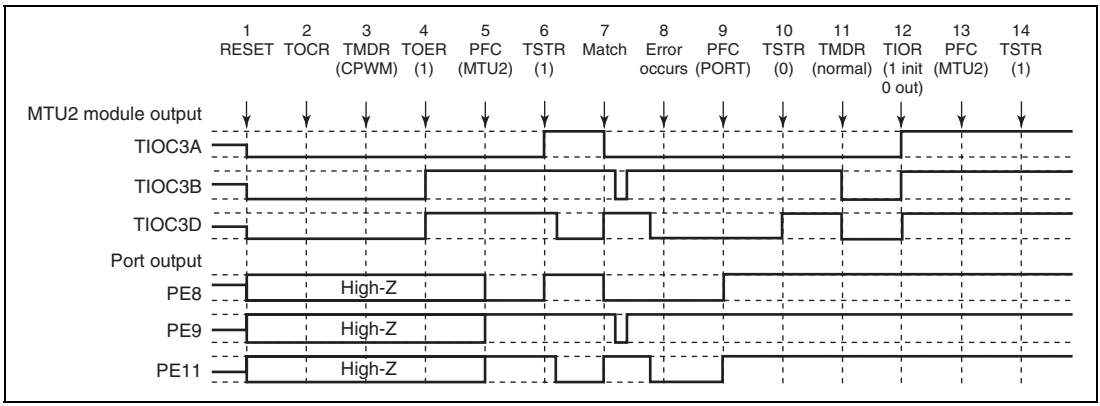



**Figure 10.158 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode**

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
3. Set complementary PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The complementary PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU2 output becomes the complementary PWM output initial value.)
11. Set normal mode. (MTU2 output goes low.)
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

**Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in PWM Mode 1:** Figure 10.159 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

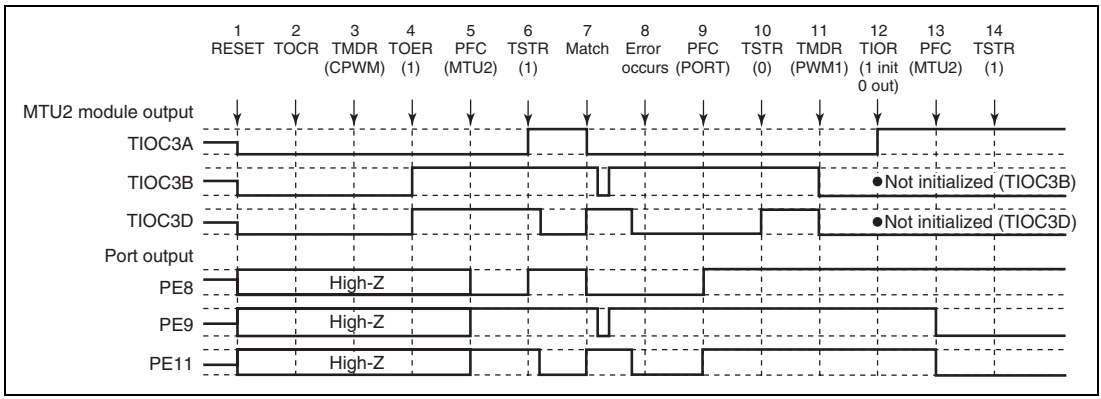



**Figure 10.159 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1**

1 to 10 are the same as in figure 10.158.

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.

**Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode:** Figure 10.160 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time the counter was stopped).

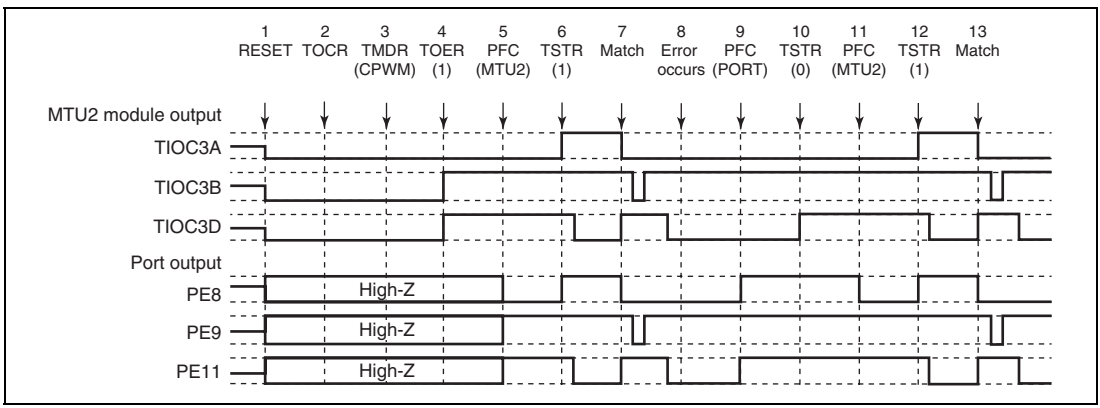



**Figure 10.160 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

1 to 10 are the same as in figure 10.158.

11. Set MTU2 output with the PFC.
12. Operation is restarted by TSTR.
13. The complementary PWM waveform is output on compare-match occurrence.

**Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode:** Figure 10.161 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using completely new cycle and duty settings).

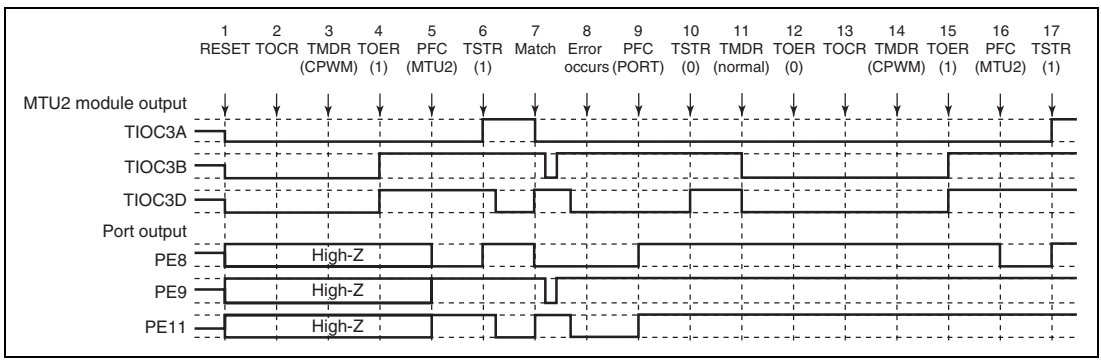



**Figure 10.161 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

1 to 10 are the same as in figure 10.158.

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.

**Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode:** Figure 10.162 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

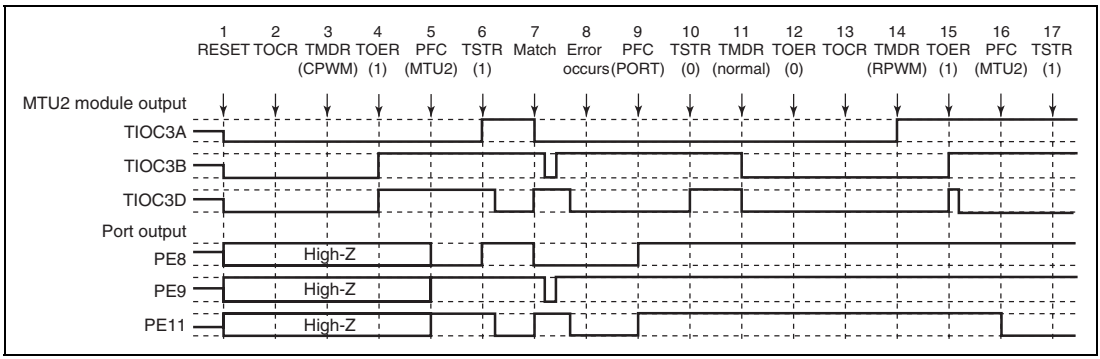



**Figure 10.162 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode**

1 to 10 are the same as in figure 10.158.

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.

**Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Normal Mode:** Figure 10.163 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

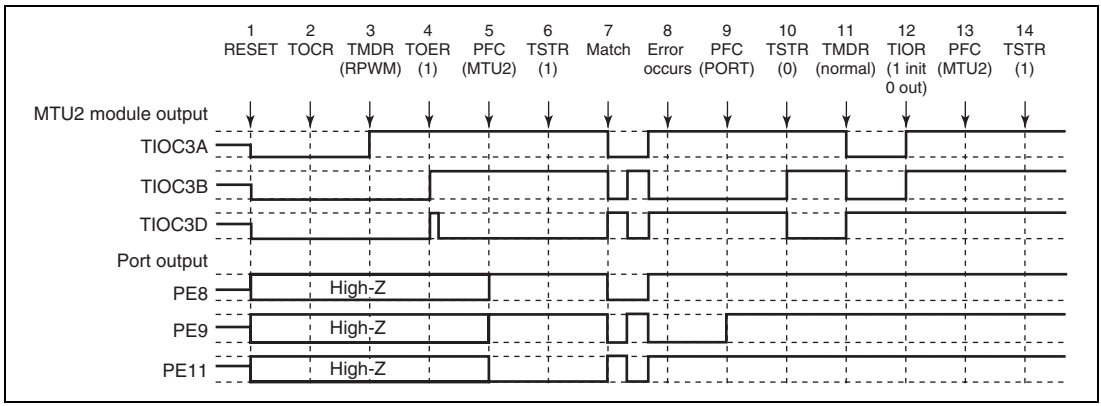



**Figure 10.163 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode**

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
3. Set reset-synchronized PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The reset-synchronized PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU2 output becomes the reset-synchronized PWM output initial value.)
11. Set normal mode. (MTU2 positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

**Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in PWM Mode 1:** Figure 10.164 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

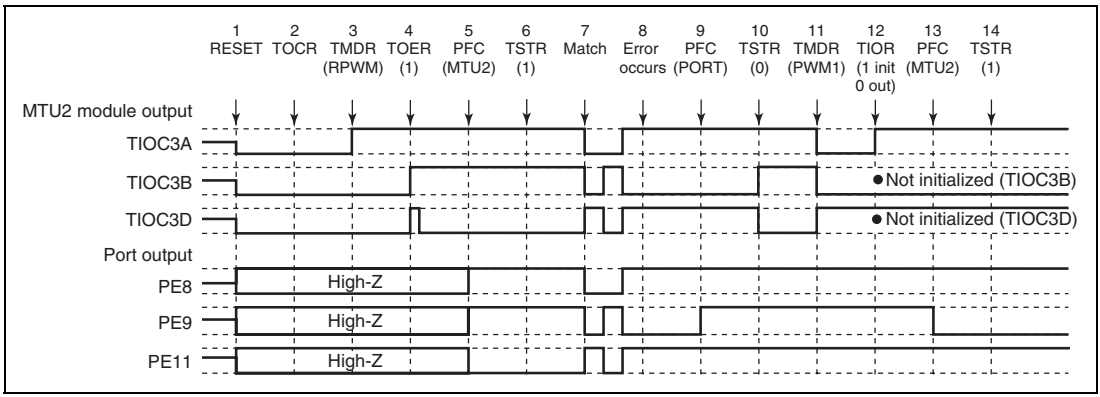



**Figure 10.164 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1**

1 to 10 are the same as in figure 10.163.

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.

**Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode:** Figure 10.165 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

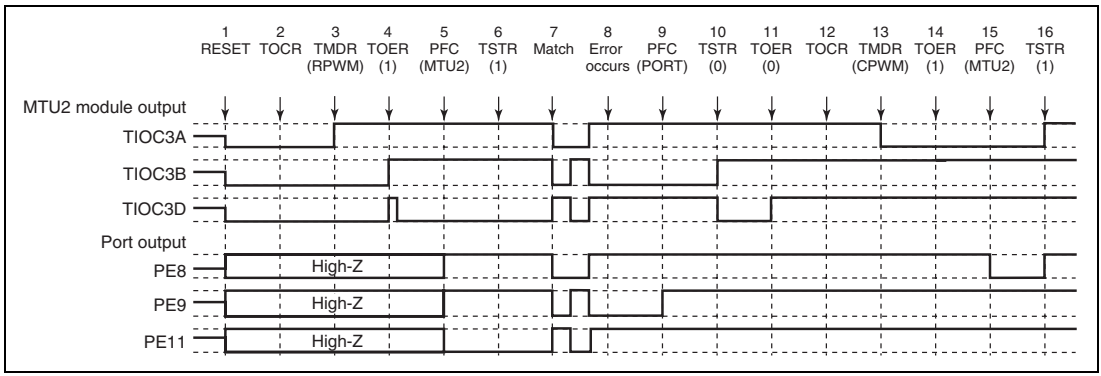



**Figure 10.165 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode**

1 to 10 are the same as in figure 10.163.

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.

**Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode:** Figure 10.166 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

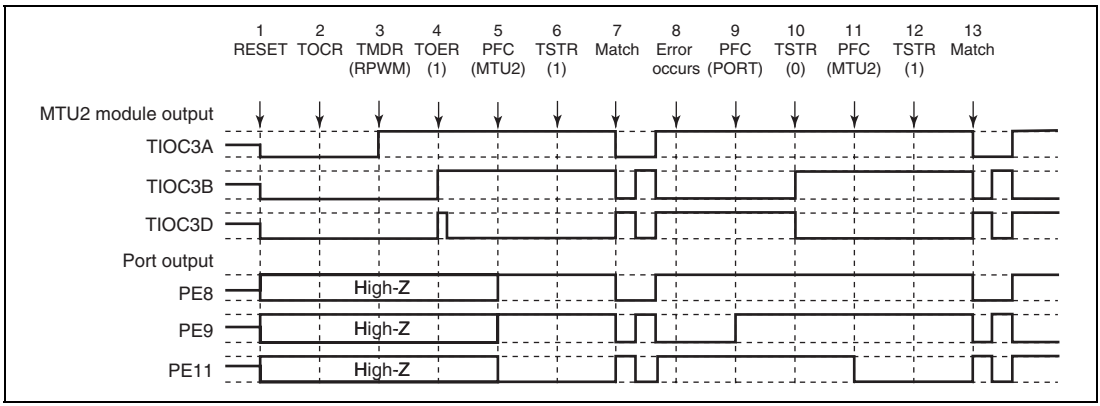



**Figure 10.166 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode**

1 to 10 are the same as in figure 10.163.

11. Set MTU2 output with the PFC.
12. Operation is restarted by TSTR.
13. The reset-synchronized PWM waveform is output on compare-match occurrence.

# Section 11 Multi-Function Timer Pulse Unit 2S (MTU2S)

This LSI has an on-chip multi-function timer pulse unit 2S (MTU2S) that comprises three 16-bit timer channels. The MTU2S includes channels 3 to 5 of the MTU2. For details, refer to section 10, Multi-Function Timer Pulse Unit 2 (MTU2). To distinguish from the MTU2, "S" is added to the end of the MTU2S input/output pin and register names. For example, TIOC3A is called TIOC3AS and TGRA_3 is called TGRA_3S in this section.

The MTU2S can operate at 80 MHz max. for complementary PWM output functions or at 40 MHz max. for the other functions.

**Table 11.1 MTU2S Functions**

| Item | | Channel 3 | Channel 4 | Channel 5 |
|---|---|---|---|---|
| Count clock | | MIϕ/1<br>MIϕ/4<br>MIϕ/16<br>MIϕ/64<br>MIϕ/256<br>MIϕ/1024 | MIϕ/1<br>MIϕ/4<br>MIϕ/16<br>MIϕ/64<br>MIϕ/256<br>MIϕ/1024 | MIϕ/1<br>MIϕ/4<br>MIϕ/16<br>MIϕ/64 |
| General registers | | TGRA_3S<br>TGRB_3S | TGRA_4S<br>TGRB_4S | TGRU_5S<br>TGRV_5S<br>TGRW_5S |
| General registers/ buffer registers | | TGRC_3S<br>TGRD_3S | TGRC_4S<br>TGRD_4S | — |
| I/O pins | | TIOC3BS<br>TIOC3DS | TIOC4AS<br>TIOC4BS<br>TIOC4CS<br>TIOC4DS | Input pins<br>TIC5US<br>TIC5VS<br>TIC5WS |
| Counter clear function | | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture |
| Compare match output | 0 output | √ | √ | — |
| | 1 output | √ | √ | — |
| | Toggle output | √ | √ | — |
| Input capture function | | √ | √ | √ |
| Synchronous operation | | √ | √ | — |
| PWM mode 1 | | √ | √ | — |
| PWM mode 2 | | — | — | — |
| Complementary PWM mode | | √ | √ | — |
| Reset PWM mode | | √ | √ | — |
| AC synchronous motor drive mode | | — | — | — |
| Phase counting mode | | — | — | — |
| Buffer operation | | √ | √ | — |

| Item | Channel 3 | Channel 4 | Channel 5 |
|---|---|---|---|
| Counter function of compensation for dead time | — | — | √ |
| DTC activation | TGR compare match or input capture | TGR compare match or input capture and TCNT overflow or underflow | TGR compare match or input capture |
| A/D converter start trigger | TGRA_3S compare match or input capture | TGRA_4S compare match or input capture<br>TCNT_4S underflow (trough) in complementary PWM mode | — |
| Interrupt sources | 5 sources<br>• Compare match or input capture 3AS<br>• Compare match or input capture 3BS<br>• Compare match or input capture 3CS<br>• Compare match or input capture 3DS<br>• Overflow | 5 sources<br>• Compare match or input capture 4AS<br>• Compare match or input capture 4BS<br>• Compare match or input capture 4CS<br>• Compare match or input capture 4DS<br>• Overflow or underflow | 3 sources<br>• Compare match or input capture 5US<br>• Compare match or input capture 5VS<br>• Compare match or input capture 5WS |
| A/D converter start request delaying function | — | • A/D converter start request at a match between TADCORA_4S and TCNT_4S<br>• A/D converter start request at a match between TADCORB_4S and TCNT_4S | — |

| Item | Channel 3 | Channel 4 | Channel 5 |
|---|---|---|---|
| Interrupt skipping function | • Skips TGRA_3S compare match interrupts | • Skips TCIV_4S interrupts | — |

[Legend]
√: Possible
—: Not possible

## 11.1 Input/Output Pins

**Table 11.2 Pin Configuration**

| Channel | Symbol | I/O | Function |
|---|---|---|---|
| 3 | TIOC3BS | I/O | TGRB_3S input capture input/output compare output/PWM output pin |
| | TIOC3DS | I/O | TGRD_3S input capture input/output compare output/PWM output pin |
| 4 | TIOC4AS | I/O | TGRA_4S input capture input/output compare output/PWM output pin |
| | TIOC4BS | I/O | TGRB_4S input capture input/output compare output/PWM output pin |
| | TIOC4CS | I/O | TGRC_4S input capture input/output compare output/PWM output pin |
| | TIOC4DS | I/O | TGRD_4S input capture input/output compare output/PWM output pin |
| 5 | TIC5US | Input | TGRU_5S input capture input/external pulse input pin |
| | TIC5VS | Input | TGRV_5S input capture input/external pulse input pin |
| | TIC5WS | Input | TGRW_5S input capture input/external pulse input pin |

## 11.2 Register Descriptions

The MTU2S has the following registers. For details on register addresses and register states during each process, refer to section 25, List of Registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 3 is expressed as TCR_3S.

**Table 11.3 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Timer control register_3S | TCR_3S | R/W | H'00 | H'FFFFC600 | 8, 16, 32 |
| Timer control register_4S | TCR_4S | R/W | H'00 | H'FFFFC601 | 8 |
| Timer mode register_3S | TMDR_3S | R/W | H'00 | H'FFFFC602 | 8, 16 |
| Timer mode register_4S | TMDR_4S | R/W | H'00 | H'FFFFC603 | 8 |
| Timer I/O control register H_3S | TIORH_3S | R/W | H'00 | H'FFFFC604 | 8, 16, 32 |
| Timer I/O control register L_3S | TIORL_3S | R/W | H'00 | H'FFFFC605 | 8 |
| Timer I/O control register H_4S | TIORH_4S | R/W | H'00 | H'FFFFC606 | 8, 16 |
| Timer I/O control register L_4S | TIORL_4S | R/W | H'00 | H'FFFFC607 | 8 |
| Timer interrupt enable register_3S | TIER_3S | R/W | H'00 | H'FFFFC608 | 8, 16 |
| Timer interrupt enable register_4S | TIER_4S | R/W | H'00 | H'FFFFC609 | 8 |
| Timer output master enable register S | TOERS | R/W | H'C0 | H'FFFFC60A | 8 |
| Timer gate control register S | TGCRS | R/W | H'80 | H'FFFFC60D | 8 |
| Timer output control register 1S | TOCR1S | R/W | H'00 | H'FFFFC60E | 8, 16 |
| Timer output control register 2S | TOCR2S | R/W | H'00 | H'FFFFC60F | 8 |
| Timer counter_3S | TCNT_3S | R/W | H'0000 | H'FFFFC610 | 16, 32 |
| Timer counter_4S | TCNT_4S | R/W | H'0000 | H'FFFFC612 | 16 |
| Timer cycle data register S | TCDRS | R/W | H'FFFF | H'FFFFC614 | 16, 32 |
| Timer dead time data register S | TDDRS | R/W | H'FFFF | H'FFFFC616 | 16 |
| Timer general register A_3S | TGRA_3S | R/W | H'FFFF | H'FFFFC618 | 16, 32 |
| Timer general register B_3S | TGRB_3S | R/W | H'FFFF | H'FFFFC61A | 16 |
| Timer general register A_4S | TGRA_4S | R/W | H'FFFF | H'FFFFC61C | 16, 32 |
| Timer general register B_4S | TGRB_4S | R/W | H'FFFF | H'FFFFC61E | 16 |

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Timer subcounter S | TCNTSS | R | H'0000 | H'FFFFC620 | 16, 32 |
| Timer cycle buffer register S | TCBRS | R/W | H'FFFF | H'FFFFC622 | 16 |
| Timer general register C_3S | TGRC_3S | R/W | H'FFFF | H'FFFFC624 | 16, 32 |
| Timer general register D_3S | TGRD_3S | R/W | H'FFFF | H'FFFFC626 | 16 |
| Timer general register C_4S | TGRC_4S | R/W | H'FFFF | H'FFFFC628 | 16, 32 |
| Timer general register D_4S | TGRD_4S | R/W | H'FFFF | H'FFFFC62A | 16 |
| Timer status register_3S | TSR_3S | R/W | H'C0 | H'FFFFC62C | 8, 16 |
| Timer status register_4S | TSR_4S | R/W | H'C0 | H'FFFFC62D | 8 |
| Timer interrupt skipping set register S | TITCRS | R/W | H'00 | H'FFFFC630 | 8, 16 |
| Timer interrupt skipping counter S | TITCNTS | R | H'00 | H'FFFFC631 | 8 |
| Timer buffer transfer set register S | TBTERS | R/W | H'00 | H'FFFFC632 | 8 |
| Timer dead time enable register S | TDERS | R/W | H'01 | H'FFFFC634 | 8 |
| Timer output level buffer register S | TOLBRS | R/W | H'00 | H'FFFFC636 | 8 |
| Timer buffer operation transfer mode register_3S | TBTM_3S | R/W | H'00 | H'FFFFC638 | 8, 16 |
| Timer buffer operation transfer mode register_4S | TBTM_4S | R/W | H'00 | H'FFFFC639 | 8 |
| Timer A/D converter start request control register S | TADCRS | R/W | H'0000 | H'FFFFC640 | 16 |
| Timer A/D converter start request cycle set register A_4S | TADCORA_4S | R/W | H'FFFF | H'FFFFC644 | 16, 32 |
| Timer A/D converter start request cycle set register B_4S | TADCORB_4S | R/W | H'FFFF | H'FFFFC646 | 16 |
| Timer A/D converter start request cycle set buffer register A_4S | TADCOBRA_4S | R/W | H'FFFF | H'FFFFC648 | 16, 32 |
| Timer A/D converter start request cycle set buffer register B_4S | TADCOBRB_4S | R/W | H'FFFF | H'FFFFC64A | 16 |

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Timer synchronous clear register S | TSYCRS | R/W | H'00 | H'FFFFC650 | 8 |
| Timer waveform control register S | TWCRS | R/W | H'00 | H'FFFFC660 | 8 |
| Timer start register S | TSTRS | R/W | H'00 | H'FFFFC680 | 8, 16 |
| Timer synchronous register S | TSYRS | R/W | H'00 | H'FFFFC681 | 8 |
| Timer read/write enable register S | TRWERS | R/W | H'01 | H'FFFFC684 | 8 |
| Timer counter U_5S | TCNTU_5S | R/W | H'0000 | H'FFFFC880 | 16, 32 |
| Timer general register U_5S | TGRU_5S | R/W | H'FFFF | H'FFFFC882 | 16 |
| Timer control register U_5S | TCRU_5S | R/W | H'00 | H'FFFFC884 | 8 |
| Timer I/O control register U_5S | TIORU_5S | R/W | H'00 | H'FFFFC886 | 8 |
| Timer counter V_5S | TCNTV_5S | R/W | H'0000 | H'FFFFC890 | 16, 32 |
| Timer general register V_5S | TGRV_5S | R/W | H'FFFF | H'FFFFC892 | 16 |
| Timer control register V_5S | TCRV_5S | R/W | H'00 | H'FFFFC894 | 8 |
| Timer I/O control register V_5S | TIORV_5S | R/W | H'00 | H'FFFFC896 | 8 |
| Timer counter W_5S | TCNTW_5S | R/W | H'0000 | H'FFFFC8A0 | 16, 32 |
| Timer general register W_5S | TGRW_5S | R/W | H'FFFF | H'FFFFC8A2 | 16 |
| Timer control register W_5S | TCRW_5S | R/W | H'00 | H'FFFFC8A4 | 8 |
| Timer I/O control register W_5S | TIORW_5S | R/W | H'00 | H'FFFFC8A6 | 8 |
| Timer status register_5S | TSR_5S | R/W | H'00 | H'FFFFC8B0 | 8 |
| Timer interrupt enable register_5S | TIER_5S | R/W | H'00 | H'FFFFC8B2 | 8 |
| Timer start register_5S | TSTR_5S | R/W | H'00 | H'FFFFC8B4 | 8 |
| Timer compare match clear register S | TCNTCMPCLRS | R/W | H'00 | H'FFFFC8B6 | 8 |

# Section 12 Port Output Enable (POE)

The port output enable (POE) module can be used to place the large current pins (pins multiplexed with TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D in the MTU2 and TIOC3BS, TIOC3DS, TIOC4AS, TIOC4BS, TIOC4CS, and TIOC4DS in the MTU2S) and the pins for channel 0 of the MTU2 (pins multiplexed with TIOC0A, TIOC0B, TIOC0C, and TIOC0D) in the high-impedance state, upon transitions on the $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, and $\overline{\text{POE8}}$ input pins, in accord with the output states of the large current pins, or by register settings. It can also generate interrupt requests at the same time.

## 12.1 Features

- Each of the $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, and $\overline{\text{POE8}}$ input pins can be set for falling edge, Pϕ/8 × 16, Pϕ/16 × 16, or Pϕ/128 × 16 low-level sampling.
- The large current pins and the pins for channel 0 of the MTU2 can be placed in the high-impedance state on the falling edge or low-level sampling of the $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, and $\overline{\text{POE8}}$ pins.
- Output levels on the large current pins are compared and if active-level outputs continue on multiple pins simultaneously for one cycle or more, the large current pins can be placed in the high-impedance state.
- The large current pins and the pins for channel 0 of the MTU2 can be placed in the high-impedance state by modifying the POE register setting.
- Interrupts can be generated by input-level sampling or output-level comparison results.

The POE has input level detection circuits, output level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in figure 12.1.

In addition to control by the POE, the large current pins can be placed in the high-impedance state when the oscillator stops or in software standby state. For details, refer to appendix A, Pin States.

Figure 12.1 shows a block diagram of the POE.

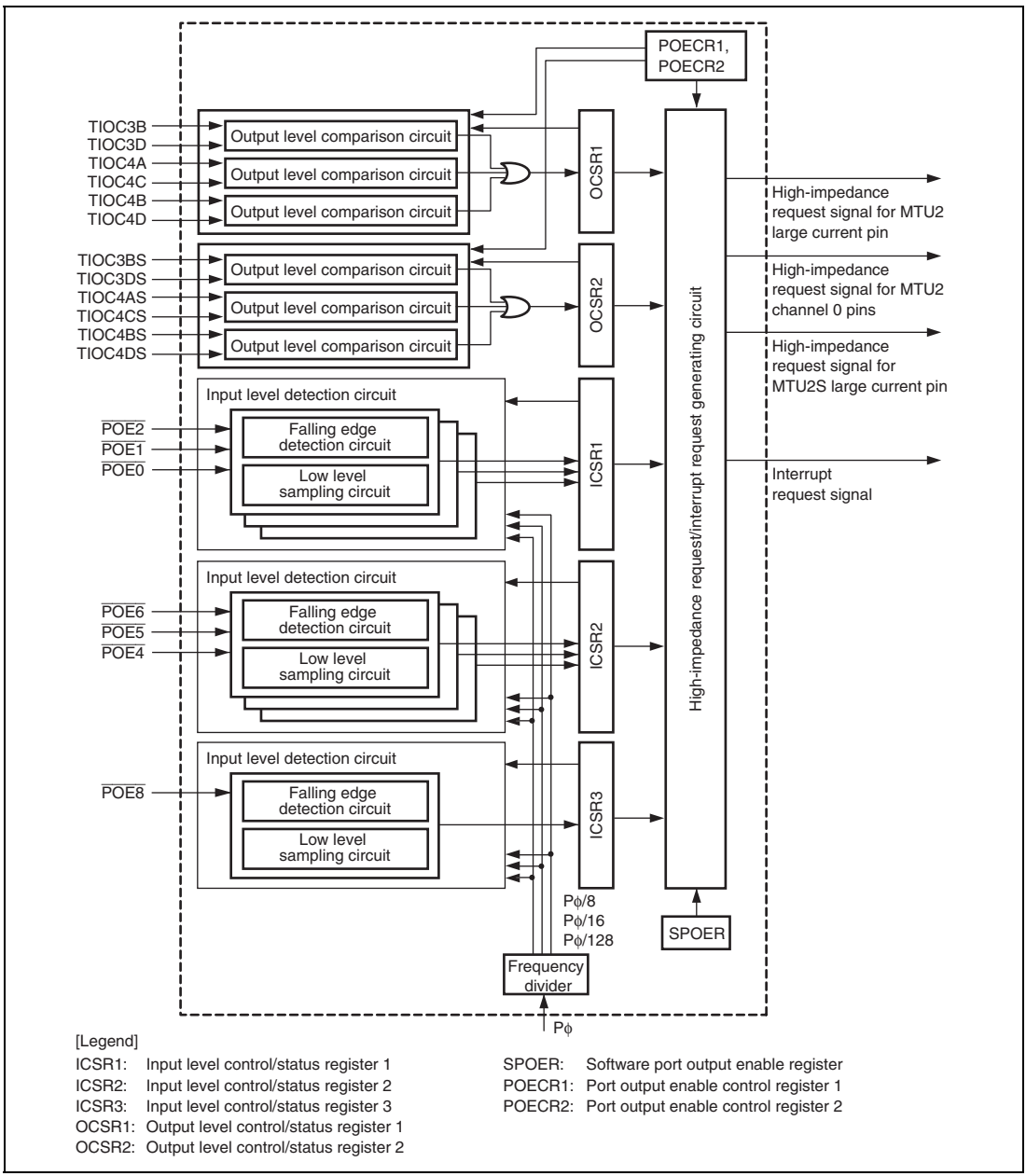



**Figure 12.1 Block Diagram of POE**

## 12.2 Input/Output Pins

### Table 12.1 Pin Configuration

| Name | Symbol | I/O | Description |
|---|---|---|---|
| Port output enable input pins 0 to 2 | $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$ | Input | Input request signals to place the large current pins for the MTU2 in the high-impedance state |
| Port output enable input pins 4 to 6 | $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$ | Input | Input request signals to place the large current pins for the MTU2S in the high-impedance state |
| Port output enable input pin 8 | $\overline{\text{POE8}}$ | Input | Inputs a request signal to place pins for channel 0 in MTU2 in the high-impedance state |

Table 12.2 shows output-level comparisons with pin combinations.

**Table 12.2 Pin Combinations**

| Pin Combination | I/O | Description |
|---|---|---|
| PE9/TIOC3B and PE11/TIOC3D<br>PE12/TIOC4A and PE14/TIOC4C<br>PE13/TIOC4B and PE15/TIOC4D | Output | The large current pins for the MTU2 are placed in the high-impedance state when the pins simultaneously output an active level (low level when the output level select P (OLSP) bit of the timer output control register (TOCR) in the MTU2 is 0 or high level when the bit is 1) for one or more cycles of the peripheral clock (Pϕ).<br>This active level comparison is done when the MTU2 output function or general output function is selected in the pin function controller. If another function is selected, the output level is not checked.<br>Pin combinations for output comparison and high-impedance control can be selected by POE registers. |
| PE16/TIOC3BS and PE17/TIOC3DS<br>PE18/TIOC4AS and PE20/TIOC4CS<br>PE19/TIOC4BS and PE21/TIOC4DS | Output | The large current pins for the MTU2S are placed in the high-impedance state when the pins simultaneously output an active level (low level when the output level select P (OLSP) bit of the timer output control register (TOCR) in the MTU2S is 0 or high level when the bit is 1) for one or more cycles of the peripheral clock (Pϕ).<br>This active level comparison is done when the MTU2S output function or general output function is selected in the pin function controller. If another function is selected, the output level is not checked.<br>Pin combinations for output comparison and high-impedance control can be selected by POE registers. |

## 12.3 Register Descriptions

The POE has the following registers. For details on register addresses and register states during each processing, refer to section 25, List of Registers.

**Table 12.3 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Input level control/status register 1 | ICSR1 | R/W | H'0000 | H'FFFFD000 | 8, 16, 32 |
| Output level control/status register 1 | OCSR1 | R/W | H'0000 | H'FFFFD002 | 8, 16 |
| Input level control/status register 2 | ICSR2 | R/W | H'0000 | H'FFFFD004 | 8, 16, 32 |
| Output level control/status register 2 | OCSR2 | R/W | H'0000 | H'FFFFD006 | 8, 16 |
| Input level control/status register 3 | ICSR3 | R/W | H'0000 | H'FFFFD008 | 8, 16 |
| Software port output enable register | SPOER | R/W | H'00 | H'FFFFD00A | 8 |
| Port output enable control register 1 | POECR1 | R/W | H'00 | H'FFFFD00B | 8 |
| Port output enable control register 2 | POECR2 | R/W | H'7700 | H'FFFFD00C | 8, 16 |

### 12.3.1 Input Level Control/Status Register 1 (ICSR1)

ICSR1 is a 16-bit readable/writable register that selects the $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$ pin input modes, controls the enable/disable of interrupts, and indicates status.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | POE2F | POE1F | POE0F | - | - | - | PIE1 | - | - | POE2M[1:0] | | POE1M[1:0] | | POE0M[1:0] | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/(W)*1 | R/(W)*1 | R/(W)*1 | R | R | R | R/W | R | R | R/W*2 | R/W*2 | R/W*2 | R/W*2 | R/W*2 | R/W*2 |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | POE2F | 0 | R/(W)*1 | POE2 Flag<br>This flag indicates that a high impedance request has been input to the $\overline{\text{POE2}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE2F after reading POE2F = 1 (when the falling edge is selected by bits 5 and 4 in ICSR1)<br>• By writing 0 to POE2F after reading POE2F = 1 after a high level input to POE2 is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 5 and 4 in ICSR1)<br>[Setting condition]<br>• When the input set by ICSR1 bits 5 and 4 occurs at the $\overline{\text{POE2}}$ pin |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 13 | POE1F | 0 | R/(W)*[1] | POE1 Flag<br>This flag indicates that a high impedance request has been input to the $\overline{\text{POE1}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE1F after reading POE1F = 1 (when the falling edge is selected by bits 3 and 2 in ICSR1)<br>• By writing 0 to POE1F after reading POE1F = 1 after a high level input to POE1 is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 3 and 2 in ICSR1)<br>[Setting condition]<br>• When the input set by ICSR1 bits 3 and 2 occurs at the $\overline{\text{POE1}}$ pin |
| 12 | POE0F | 0 | R/(W)*[1] | POE0 Flag<br>This flag indicates that a high impedance request has been input to the $\overline{\text{POE0}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE0F after reading POE0F = 1 (when the falling edge is selected by bits 1 and 0 in ICSR1)<br>• By writing 0 to POE0F after reading POE0F = 1 after a high level input to POE0 is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 1 and 0 in ICSR1)<br>[Setting condition]<br>• When the input set by ICSR1 bits 1 and 0 occurs at the $\overline{\text{POE0}}$ pin |
| 11 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 8 | PIE1 | 0 | R/W | Port Interrupt Enable 1<br>This bit enables/disables interrupt requests when any one of the POE0F to POE2F bits of the ICSR1 is set to 1.<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5, 4 | POE2M[1:0] | 00 | R/W*[2] | POE2 mode 1, 0<br>These bits select the input mode of the $\overline{\text{POE2}}$ pin.<br>00: Accept request on falling edge of POE2 input<br>01: Accept request when POE2 input has been sampled for 16 Pϕ/8 clock pulses and all are low level.<br>10: Accept request when POE2 input has been sampled for 16 Pϕ/16 clock pulses and all are low level.<br>11: Accept request when POE2 input has been sampled for 16 Pϕ/128 clock pulses and all are low level. |
| 3, 2 | POE1M[1:0] | 00 | R/W*[2] | POE1 mode 1, 0<br>These bits select the input mode of the $\overline{\text{POE1}}$ pin.<br>00: Accept request on falling edge of POE1 input<br>01: Accept request when POE1 input has been sampled for 16 Pϕ/8 clock pulses and all are low level.<br>10: Accept request when POE1 input has been sampled for 16 Pϕ/16 clock pulses and all are low level.<br>11: Accept request when POE1 input has been sampled for 16 Pϕ/128 clock pulses and all are low level. |
| 1, 0 | POE0M[1:0] | 00 | R/W*[2] | POE0 mode 1, 0<br>These bits select the input mode of the $\overline{\text{POE0}}$ pin.<br>00: Accept request on falling edge of POE0 input<br>01: Accept request when POE0 input has been sampled for 16 Pϕ/8 clock pulses and all are low level.<br>10: Accept request when POE0 input has been sampled for 16 Pϕ/16 clock pulses and all are low level.<br>11: Accept request when POE0 input has been sampled for 16 Pϕ/128 clock pulses and all are low level. |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

### 12.3.2 Output Level Control/Status Register 1 (OCSR1)

OCSR1 is a 16-bit readable/writable register that controls the enable/disable of both output level comparison and interrupts, and indicates status.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OSF1 | - | - | - | - | - | OCE1 | OIE1 | - | - | - | - | - | - | - | - |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/(W)*1 | R | R | R | R | R | R/W*2 | R/W | R | R | R | R | R | R | R | R |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 15 | OSF1 | 0 | R/(W)*1 | Output Short Flag 1<br>This flag indicates that any one of the three pairs of MTU2 2-phase outputs to be compared has simultaneously become an active level.<br>[Clearing condition]<br>• By writing 0 to OSF1 after reading OSF1 = 1<br>[Setting condition]<br>• When any one of the three pairs of 2-phase outputs has simultaneously become an active level |
| 14 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 9 | OCE1 | 0 | R/W*2 | Output Short High-Impedance Enable 1<br>This bit specifies whether to place the pins in the high-impedance state when the OSF1 bit in OCSR1 is set to 1.<br>0: Does not place the pins in the high-impedance state<br>1: Places the pins in the high-impedance state |
| 8 | OIE1 | 0 | R/W | Output Short Interrupt Enable 1<br>This bit enables or disables interrupt requests when the OSF1 bit in OCSR is set to 1.<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

### 12.3.3 Input Level Control/Status Register 2 (ICSR2)

ICSR2 is a 16-bit readable/writable register that selects the $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$ pin input modes, controls the enable/disable of interrupts, and indicates status.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | POE6F | POE5F | POE4F | - | - | - | PIE2 | - | - | POE6M[1:0] | | POE5M[1:0] | | POE4M[1:0] | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/(W)*1 | R/(W)*1 | R/(W)*1 | R | R | R | R/W | R | R | R/W*2 | R/W*2 | R/W*2 | R/W*2 | R/W*2 | R/W*2 |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | POE6F | 0 | R/(W)*1 | POE6 Flag<br>This flag indicates that a high impedance request has been input to the $\overline{\text{POE6}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE6F after reading POE6F = 1 (when the falling edge is selected by bits 5 and 4 in ICSR2)<br>• By writing 0 to POE6F after reading POE6F = 1 after a high level input to POE6 is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 5 and 4 in ICSR2)<br>[Setting condition]<br>• When the input condition set by bits 5 and 4 in ICSR2 occurs at the $\overline{\text{POE6}}$ pin |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 13 | POE5F | 0 | R/(W)*[1] | POE5 Flag<br>This flag indicates that a high impedance request has been input to the $\overline{\text{POE5}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE5F after reading POE5F = 1 (when the falling edge is selected by bits 3 and 2 in ICSR2)<br>• By writing 0 to POE5F after reading POE5F = 1 after a high level input to POE5 is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 3 and 2 in ICSR2)<br>[Setting condition]<br>• When the input condition set by bits 3 and 2 in ICSR2 occurs at the $\overline{\text{POE5}}$ pin |
| 12 | POE4F | 0 | R/(W)*[1] | POE4 Flag<br>This flag indicates that a high impedance request has been input to the $\overline{\text{POE4}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE4F after reading POE4F = 1 (when the falling edge is selected by bits 1 and 0 in ICSR2)<br>• By writing 0 to POE4F after reading POE4F = 1 after a high level input to POE4 is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 1 and 0 in ICSR2)<br>[Setting condition]<br>• When the input condition set by bits 1 and 0 in ICSR2 occurs at the $\overline{\text{POE4}}$ pin |
| 11 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 8 | PIE2 | 0 | R/W | Port Interrupt Enable 2<br>This bit enables/disables interrupt requests when any one of the POE4F to POE6F bits of the ICSR2 is set to 1.<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5, 4 | POE6M[1:0] | 00 | R/W*[2] | POE6 mode 1 and 0<br>These bits select the input mode of the $\overline{\text{POE6}}$ pin.<br>00: Accept request on falling edge of POE6 input<br>01: Accept request when POE6 input has been sampled for 16 Pϕ/8 clock pulses and all are at a low level.<br>10: Accept request when POE6 input has been sampled for 16 Pϕ/16 clock pulses and all are at a low level.<br>11: Accept request when POE6 input has been sampled for 16 Pϕ/128 clock pulses and all are at a low level. |
| 3, 2 | POE5M[1:0] | 00 | R/W*[2] | POE5 mode 1 and 0<br>These bits select the input mode of the $\overline{\text{POE5}}$ pin.<br>00: Accept request on falling edge of POE5 input<br>01: Accept request when POE5 input has been sampled for 16 Pϕ/8 clock pulses and all are at a low level.<br>10: Accept request when POE5 input has been sampled for 16 Pϕ/16 clock pulses and all are at a low level.<br>11: Accept request when POE5 input has been sampled for 16 Pϕ/128 clock pulses and all are at a low level. |
| 1, 0 | POE4M[1:0] | 00 | R/W*[2] | POE4 mode 1 and 0<br>These bits select the input mode of the $\overline{\text{POE4}}$ pin.<br>00: Accept request on falling edge of POE4 input<br>01: Accept request when POE4 input has been sampled for 16 Pϕ/8 clock pulses and all are at a low level.<br>10: Accept request when POE4 input has been sampled for 16 Pϕ/16 clock pulses and all are at a low level.<br>11: Accept request when POE4 input has been sampled for 16 Pϕ/128 clock pulses and all are at a low level. |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

### 12.3.4 Output Level Control/Status Register 2 (OCSR2)

OCSR2 is a 16-bit readable/writable register that controls the enable/disable of both output level comparison and interrupts, and indicates status.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OSF2 | - | - | - | - | - | OCE2 | OIE2 | - | - | - | - | - | - | - | - |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/(W)*1 | R | R | R | R | R | R/W*2 | R/W | R | R | R | R | R | R | R | R |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 15 | OSF2 | 0 | R/(W)*1 | Output Short Flag 2<br>This flag indicates that any one of the three pairs of MTU2S 2-phase outputs to be compared has simultaneously become an active level.<br>[Clearing condition]<br>• By writing 0 to OSF2 after reading OSF2 = 1<br>[Setting condition]<br>• When any one of the three pairs of 2-phase outputs has simultaneously become an active level |
| 14 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 9 | OCE2 | 0 | R/W*2 | Output Short High-Impedance Enable 2<br>This bit specifies whether to place the pins in the high-impedance state when the OSF2 bit in OCSR2 is set to 1.<br>0: Does not place the pins in the high-impedance state<br>1: Places the pins in the high-impedance state |
| 8 | OIE2 | 0 | R/W | Output Short Interrupt Enable 2<br>This bit enables or disables interrupt requests when the OSF2 bit in OCSR2 is set to 1.<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

### 12.3.5 Input Level Control/Status Register 3 (ICSR3)

ICSR3 is a 16-bit readable/writable register that selects the $\overline{\text{POE8}}$ pin input mode, controls the enable/disable of interrupts, and indicates status.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | POE8F | - | - | POE8E | PIE3 | - | - | - | - | - | - | POE8M[1:0] | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R/(W)*1 | R | R | R/W*2 | R/W | R | R | R | R | R | R | R/W*2 | R/W*2 |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 15 to 13 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 12 | POE8F | 0 | R/(W)*1 | POE8 Flag<br>This flag indicates that a high impedance request has been input to the $\overline{\text{POE8}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE8F after reading POE8F = 1 (when the falling edge is selected by bits 1 and 0 in ICSR3)<br>• By writing 0 to POE8F after reading POE8F = 1 after a high level input to POE8 is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 1 and 0 in ICSR3)<br>[Setting condition]<br>• When the input condition set by bits 1 and 0 in ICSR3 occurs at the $\overline{\text{POE8}}$ pin |
| 11, 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 9 | POE8E | 0 | R/W*[2] | POE8 High-Impedance Enable<br>This bit specifies whether to place the pins in the high-impedance state when the POE8F bit in ICSR3 is set to 1.<br>0: Does not place the pins in the high-impedance state<br>1: Places the pins in the high-impedance state |
| 8 | PIE3 | 0 | R/W | Port Interrupt Enable 3<br>This bit enables or disables interrupt requests when the POE8 bit in ICSR3 is set to 1.<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |
| 7 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1, 0 | POE8M[1:0] | 00 | R/W*[2] | POE8 mode 1 and 0<br>These bits select the input mode of the $\overline{\text{POE8}}$ pin.<br>00: Accept request on falling edge of POE8 input<br>01: Accept request when POE8 input has been sampled for 16 Pϕ/8 clock pulses and all are low level.<br>10: Accept request when POE8 input has been sampled for 16 Pϕ/16 clock pulses and all are low level.<br>11: Accept request when POE8 input has been sampled for 16 Pϕ/128 clock pulses and all are low level. |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Can be modified only once after a power-on reset.

### 12.3.6 Software Port Output Enable Register (SPOER)

SPOER is an 8-bit readable/writable register that controls high-impedance state of the pins.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | MTU2S HIZ | MTU2 CH0HIZ | MTU2 CH34HIZ |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 2 | MTU2SHIZ | 0 | R/W | MTU2S Output High-Impedance<br>This bit specifies whether to place the large current pins for the MTU2S in the high-impedance state.<br>0: Does not place the pins in the high-impedance state<br>[Clearing conditions]<br>• Power-on reset<br>• By writing 0 to MTU2SHIZ after reading MTU2SHIZ = 1<br>1: Places the pins in the high-impedance state<br>[Setting condition]<br>• By writing 1 to MTU2SHIZ |
| 1 | MTU2CH0HIZ | 0 | R/W | MTU2 Channel 0 Output High-Impedance<br>This bit specifies whether to place the pins for channel 0 in the MTU2 in the high-impedance state.<br>0: Does not place the pins in the high-impedance state<br>[Clearing conditions]<br>• Power-on reset<br>• By writing 0 to MTU2CH0HIZ after reading MTU2CH0HIZ = 1<br>1: Places the pins in the high-impedance state<br>[Setting condition]<br>• By writing 1 to MTU2CH0HIZ |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 0 | MTU2CH34HIZ | 0 | R/W | MTU2 Channel 3 and 4 Output High-Impedance<br>This bit specifies whether to place the large current pins for the MTU2 in the high-impedance state.<br>0: Does not place the pins in the high-impedance state<br>[Clearing conditions]<br>• Power-on reset<br>• By writing 0 to MTU2CH34HIZ after reading MTU2CH34HIZ = 1<br>1: Places the pins in the high-impedance state<br>[Setting condition]<br>• By writing 1 to MTU2CH34HIZ |

### 12.3.7 Port Output Enable Control Register 1 (POECR1)

POECR1 is an 8-bit readable/writable register that controls high-impedance state of the pins.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | MTU2 PE3ZE | MTU2 PE2ZE | MTU2 PE1ZE | MTU2 PE0ZE |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R/W* | R/W* | R/W* | R/W* |

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

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 to 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 3 | MTU2PE3ZE | 0 | R/W* | MTU2 PE3 High-Impedance Enable<br>This bit specifies whether to place the PE3/TIOC0D pin for channel 0 in the MTU2 in the high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1.<br>0: Does not place the pin in the high-impedance state<br>1: Places the pin in the high-impedance state |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 2 | MTU2PE2ZE | 0 | R/W* | MTU2 PE2 High-Impedance Enable<br>This bit specifies whether to place the PE2/TIOC0C pin for channel 0 in the MTU2 in the high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1.<br>0: Does not place the pin in the high-impedance state<br>1: Places the pin in the high-impedance state |
| 1 | MTU2PE1ZE | 0 | R/W* | MTU2 PE1 High-Impedance Enable<br>This bit specifies whether to place the PE1/TIOC0B pin for channel 0 in the MTU2 in the high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1.<br>0: Does not place the pin in the high-impedance state<br>1: Places the pin in the high-impedance state |
| 0 | MTU2PE0ZE | 0 | R/W* | MTU2 PE0 High-Impedance Enable<br>This bit specifies whether to place the PE0/TIOC0A pin for channel 0 in the MTU2 in the high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1.<br>0: Does not place the pin in the high-impedance state<br>1: Places the pin in the high-impedance state |

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

### 12.3.8 Port Output Enable Control Register 2 (POECR2)

POECR2 is a 16-bit readable/writable register that controls high-impedance state of the pins.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | MTU2 P1CZE | MTU2 P2CZE | MTU2 P3CZE | - | MTU2S P1CZE | MTU2S P2CZE | MTU2S P3CZE | - | - | - | - | - | - | - | - |
| Initial value: | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W* | R/W* | R/W* | R | R/W* | R/W* | R/W* | R | R | R | R | R | R | R | R |

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

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | MTU2P1CZE | 1 | R/W* | MTU2 Port 1 Output Comparison/High-Impedance Enable<br>This bit specifies whether to compare output levels on the large current pins for the MTU2, PE9/TIOC3B and PE11/TIOC3D, and to place them in the high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F, POE1F, POE2F, and MTU2CH34HIZ bits is set to 1.<br>0: Does not compare output levels or place the pins in the high-impedance state<br>1: Compares output levels and places the pins in the high-impedance state |
| 13 | MTU2P2CZE | 1 | R/W* | MTU2 Port 2 Output Comparison/High-Impedance Enable<br>This bit specifies whether to compare output levels on the large current pins for the MTU2, PE12/TIOC4A and PE14/TIOC4C, and to place them in the high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F, POE1F, POE2F, and MTU2CH34HIZ bits is set to 1.<br>0: Does not compare output levels or place the pins in the high-impedance state<br>1: Compares output levels and places the pins in the high-impedance state |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 12 | MTU2P3CZE | 1 | R/W* | MTU2 Port 3 Output Comparison/High-Impedance Enable<br>This bit specifies whether to compare output levels on the large current pins for the MTU2S, PE13/TIOC4B and PE15/TIOC4D, and to place them in the high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F, POE1F, POE2F, and MTU2CH34HIZ bits is set to 1.<br>0: Does not compare output levels or place the pins in the high-impedance state<br>1: Compares output levels and places the pins in the high-impedance state |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | MTU2SP1CZE | 1 | R/W* | MTU2S Port 1 Output Comparison/High-Impedance Enable<br>This bit specifies whether to compare output levels on the large current pins for the MTU2S, PE16/TIOC3BS and PE17/TIOC3DS, and to place them in the high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when any one of the POE4F, POE5F, POE6F, and MTU2SHIZ bits is set to 1.<br>0: Does not compare output levels or place the pins in the high-impedance state<br>1: Compares output levels and places the pins in the high-impedance state |
| 9 | MTU2SP2CZE | 1 | R/W* | MTU2S Port 2 Output Comparison/High-Impedance Enable<br>This bit specifies whether to compare output levels on the large current pins for the MTU2S, PE18/TIOC4AS and PE20/TIOC4CS, and to place them in the high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when any one of the POE4F, POE5F, POE6F, and MTU2SHIZ bits is set to 1.<br>0: Does not compare output levels or place the pins in the high-impedance state<br>1: Compares output levels and places the pins in the high-impedance state |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 8 | MTU2SP3CZE | 1 | R/W* | MTU2S Port 3 Output Comparison/High-Impedance Enable<br>This bit specifies whether to compare output levels on the large current pins for the MTU2S, PE19/TIOC4BS and PE21/TIOC4DS, and to place them in the high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when any one of the POE4F, POE5F, POE6F, and MTU2SHIZ bits is set to 1.<br>0: Does not compare output levels or place the pins in the high-impedance state<br>1: Compares output levels and places the pins in the high-impedance state |
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

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

## 12.4 Operation

Table 12.4 shows the target pins for high-impedance control and conditions to place the pins in the high-impedance state.

**Table 12.4 Target Pins and Conditions for High-Impedance Control**

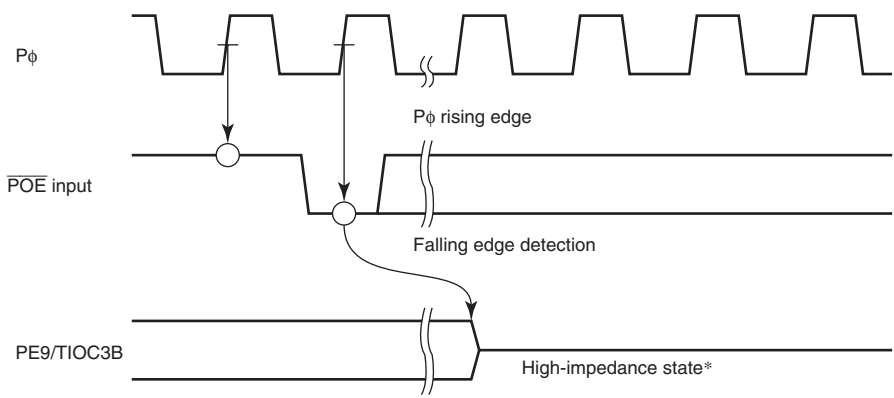
| Pins | Conditions | Detailed Conditions |
|---|---|---|
| the large current pins for the MTU2 (PE9/TIOC3B and PE11/TIOC3D) | Input level detection, output level comparison, or SPOER setting | MTU2P1CZE • ((POE2F + POE1F + POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ)) |
| the large current pins for the MTU2 (PE12/TIOC4A and PE14/TIOC4C) | Input level detection, output level comparison, or SPOER setting | MTU2P2CZE • ((POE2F + POE1F + POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ)) |
| the large current pins for the MTU2 (PE13/TIOC4B and PE15/TIOC4D) | Input level detection, output level comparison, or SPOER setting | MTU2P3CZE • ((POE2F + POE1F + POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ)) |
| the large current pins for the MTU2S (PE16/TIOC3BS and PE17/TIOC3DS) | Input level detection, output level comparison, or SPOER setting | MTU2SP1CZE • ((POE4F + POE5F + POE6F) + (OSF2 • OCE2) + (MTU2SHIZ)) |
| the large current pins for the MTU2S (PE18/TIOC4AS and PE20/TIOC4CS) | Input level detection, output level comparison, or SPOER setting | MTU2SP2CZE • ((POE4F + POE5F + POE6F) + (OSF2 • OCE2) + (MTU2SHIZ)) |
| the large current pins for the MTU2S (PE19/TIOC4BS and PE21/TIOC4DS) | Input level detection, output level comparison, or SPOER setting | MTU2SP3CZE • ((POE4F + POE5F + POE6F) + (OSF2 • OCE2) + (MTU2SHIZ)) |
| MTU2 channel 0 pin (PE0/TIOC0A) | Input level detection or SPOER setting | MTU2PE0ZE ((POE8F • POE8E) + (MTU2CH0HIZ)) |
| MTU2 channel 0 pin (PE1/TIOC0B) | Input level detection or SPOER setting | MTU2PE1ZE ((POE8F • POE8E) + (MTU2CH0HIZ)) |
| MTU2 channel 0 pin (PE2/TIOC0C) | Input level detection or SPOER setting | MTU2PE2ZE ((POE8F • POE8E) + (MTU2CH0HIZ)) |
| MTU2 channel 0 pin (PE3/TIOC0D) | Input level detection or SPOER setting | MTU2PE3ZE ((POE8F • POE8E) + (MTU2CH0HIZ)) |

### 12.4.1 Input Level Detection Operation

If the input conditions set by ICSR1 to ICSR3 occur on the $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, and $\overline{\text{POE8}}$ pins, the large current pins and the pins for channel 0 of the MTU2 are placed in the high-impedance state. Note however, that these large current pins and MTU2 pins enter high-impedance state only when general input/output function, MTU2 function, or MTU2S function is selected for these pins.

#### (1) Falling Edge Detection

When a change from a high to low level is input to the $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, and $\overline{\text{POE8}}$ pins, the large current pins and the pins for channel 0 of the MTU2 are placed in the high-impedance state. Figure 12.2 shows a sample timing after the level changes in input to the $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, and $\overline{\text{POE8}}$ pins until the respective pins enter high-impedance state.




Note: * Other large current pins also enter the high-impedance state with the same timing.

**Figure 12.2 Falling Edge Detection**

### (2) Low-Level Detection

Figure 12.3 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock selected by ICSR1 to ICSR3. If even one high level is detected during this interval, the low level is not accepted.

The timing when the large current pins enter the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.

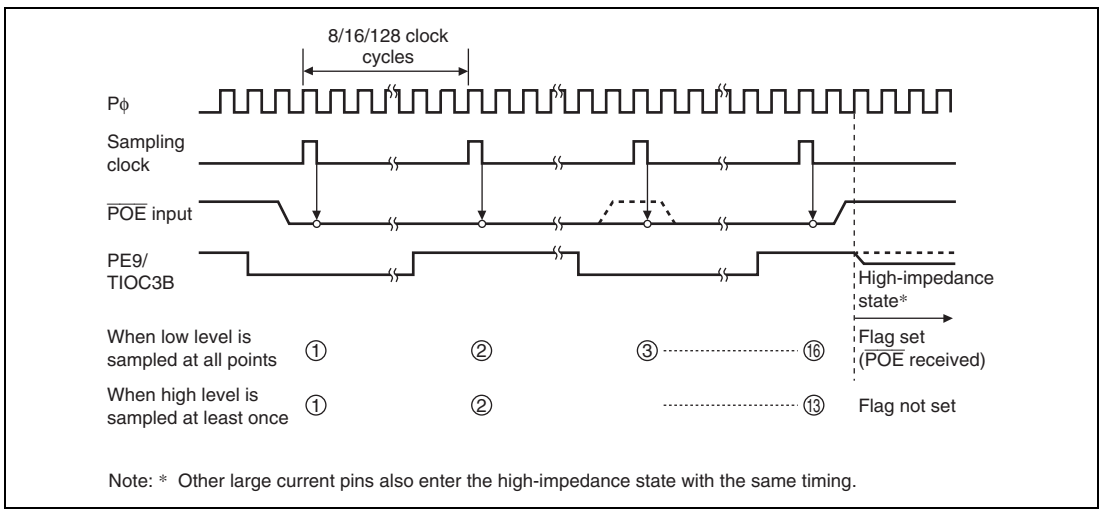



**Figure 12.3 Low-Level Detection Operation**

### 12.4.2 Output-Level Compare Operation

Figure 12.4 shows an example of the output-level compare operation for the combination of TIOC3B and TIOC3D. The operation is the same for the other pin combinations.

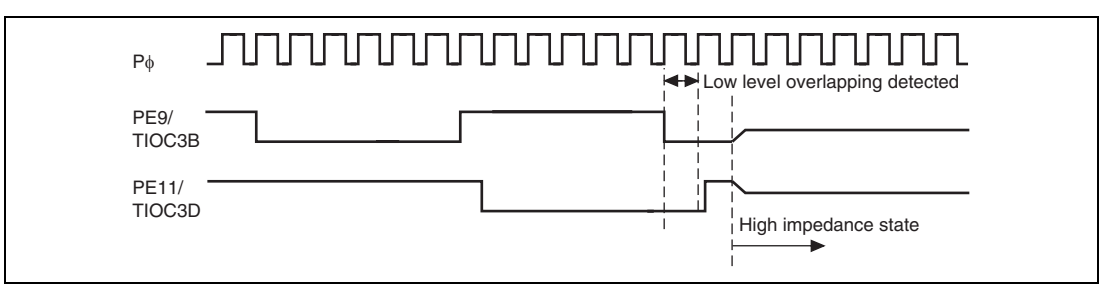



**Figure 12.4 Output-Level Compare Operation**

### 12.4.3 Release from High-Impedance State

The large current pins that have entered high-impedance state due to input-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing all of the flags in bits 12 to 15 (POE0F to POE2F, POE4F to POE6F, and POE8F) of ICSR1 to ICSR3. However, note that when low-level sampling is selected by bits 0 to 7 in ICSR1 to ICSR3, just writing 0 to a flag is ignored (the flag is not cleared); flags can be cleared by writing 0 to it only after a high level is input to the POE pin and is sampled.

The large current pins that have entered high-impedance state due to output-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing the flag in bit 15 (OCF1 and OCF2) in OCSR1 and OCSR2. However, note that just writing 0 to a flag is ignored (the flag is not cleared); flags can be cleared only after an inactive level is output from the large current pins. Inactive-level outputs can be obtained by setting the MTU2 and MTU2S internal registers.

## 12.5 Interrupts

The POE issues a request to generate an interrupt when the specified condition is satisfied during input level detection or output level comparison. Table 12.5 shows the interrupt sources and their conditions.

**Table 12.5 Interrupt Sources and Conditions**

| Name | Interrupt Source | Interrupt Flag | Condition |
|---|---|---|---|
| OEI1 | Output enable interrupt 1 | POE2F, POE1F, POE0F, and OSF1 | PIE1 • (POE2F + POE1F + POE0F) + OIE1 • OSF1 |
| OEI2 | Output enable interrupt 2 | POE8F | PIE3 • POE8F |
| OEI3 | Output enable interrupt 3 | POE4F, POE5F, POE6F, and OSF2 | PIE2 • (POE4F + POE5F + POE6F) + OIE2 • OSF2 |

## 12.6 Usage Note

### 12.6.1 Pin State when a Power-On Reset is Issued from the Watchdog Timer

When a power-on reset is issued from the watchdog timer (WDT), initialization of the pin function controller (PFC) sets initial values that select the general input function for the I/O ports. However, when a power-on reset is issued from the WDT while a pin is being handled as high impedance by the port output enable (POE), the pin is placed in the output state for one cycle of the peripheral clock (Pf), after which the function is switched to general input.

This also occurs when a power-on reset is issued from the WDT for pins that are being handled as high impedance due to short-circuit detection by the MTU2 and MTU2S.

Figure 12.5 shows the state of a pin for which the POE input has selected high impedance handling with the timer output selected when a power-on reset is issued from the WDT.

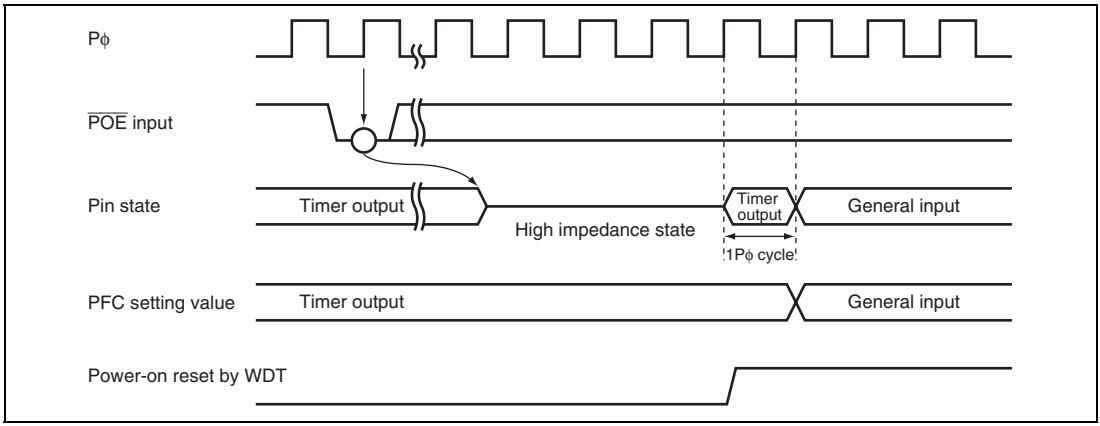



**Figure 12.5 Pin State when a Power-On Reset is Issued from the Watchdog Timer**

# Section 13 Watchdog Timer (WDT)

This LSI includes the watchdog timer (WDT).

This LSI can be reset by the overflow of the counter when the value of the counter has not been updated because of a system runaway.

The watchdog timer (WDT) is a single-channel timer that uses a peripheral clock as an input and counts the clock settling time when revoking software standby mode. It can also be used as an interval timer.

## 13.1 Features

- Can be used to ensure the clock settling time: Use the WDT to revoke software standby mode.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode: Internal resets occur after counter overflow.
- An interrupt is generated in interval timer mode
  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks
  Eight clocks (×1 to ×1/4096) that are obtained by dividing the peripheral clock can be chosen.
- Choice of two resets
  Power-on reset and manual reset are available.

Figure 13.1 shows a block diagram of the WDT.

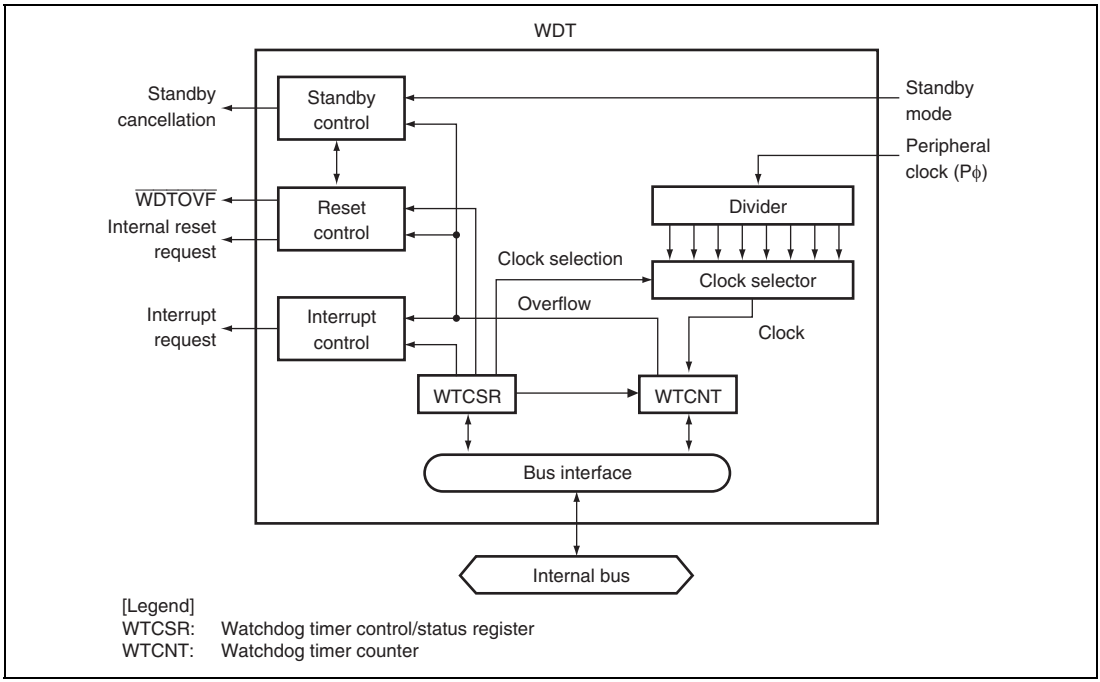



**Figure 13.1 Block Diagram of WDT**

## 13.2 Input/Output Pin for WDT

Table 13.1 lists the WDT pin configuration.

**Table 13.1 WDT Pin Configuration**

| Pin Name | Abbreviation | I/O | Description |
|---|---|---|---|
| Watchdog timer overflow | $\overline{\text{WDTOVF}}$ | Output | When an overflow occurs in watchdog timer mode, an internal reset is generated and this pin outputs the low level for one clock cycle specified by the CKS2 to CKS0 bits in WTCSR. |

## 13.3 Register Descriptions

The WDT has the following two registers. Refer to section 25, List of Registers, for the details of the addresses of these registers and the state of registers in each operating mode.

**Table 13.2 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Watchdog timer counter | WTCNT | R/W | H'00 | H'FFFFE810 | 8, 16 |
| Watchdog timer control/status register | WTCSR | R/W | H'00 | H'FFFFE812 | 8, 16 |

### 13.3.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable register that increments on the selected clock. When an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval time mode. The WTCNT counter is not initialized by an internal reset due to the WDT overflow. The WTCNT counter is initialized to H'00 only by a power-on reset using the $\overline{\text{RES}}$ pin. Use a word access to write to the WTCNT counter, with H'5A in the upper byte. Use a byte access to read WTCNT.

Note: WTCNT differs from other registers in that it is more difficult to write to. See section 13.3.3, 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 |

### 13.3.2 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, bits to select the timer mode, and overflow flags. WTCSR holds its value in an internal reset due to the WDT overflow. WTCSR is initialized to H'00 only by a power-on reset using the $\overline{\text{RES}}$ pin.

When used to count the clock settling time for revoking a software standby, it retains its value after counter overflow. Use a word access to write to WTCSR, with H'A5 in the upper byte. Use a byte access to read WTCSR.

Note: WTCSR differs from other registers in that it is more difficult to write to. See section 13.3.3, Notes on Register Access, for details.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TME | WT/IT | RSTS | WOVF | IOVF | CKS[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 | TME | 0 | R/W | Timer Enable<br>Starts and stops timer operation. Clear this bit to 0 when using the WDT to revoke software standby mode.<br>0: Timer disabled: Count-up stops and WTCNT value is retained<br>1: Timer enabled |
| 6 | WT/IT | 0 | R/W | Timer Mode Select<br>Selects whether to use the WDT as a watchdog timer or an interval timer.<br>0: Interval timer mode<br>1: Watchdog timer mode<br>Note: If WT/IT is modified when the WDT is operating, the up-count may not be performed correctly. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | RSTS | 0 | R/W | Reset Select<br>Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored.<br>0: Power-on reset<br>1: Manual reset |
| 4 | WOVF | 0 | R/W | Watchdog Timer Overflow<br>Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode.<br>0: No overflow<br>1: WTCNT has overflowed in watchdog timer mode |
| 3 | IOVF | 0 | R/W | Interval Timer Overflow<br>Indicates that the WTCNT has overflowed in interval timer mode. This bit is not set in watchdog timer mode.<br>0: No overflow<br>1: WTCNT has overflowed in interval timer mode |
| 2 to 0 | CKS[2:0] | 000 | R/W | Clock Select 2 to 0<br>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 40 MHz.<br>000: Pϕ (6.4 μs)<br>001: Pϕ /4 (25.6 μs)<br>010: Pϕ /16 (102.4 μs)<br>011: Pϕ /32 (204.8 μs)<br>100: Pϕ /64 (409.6 μs)<br>101: Pϕ /256 (1.64 ms)<br>110: Pϕ /1024 (6.55 ms)<br>111: Pϕ /4096 (26.21 ms)<br>Note: If bits CKS2 to CKS0 are modified when the WDT is operating, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not operating. |

### 13.3.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTCSR) are more difficult to write to than other registers. The procedure for writing to these registers is given below.

**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 13.2. 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.

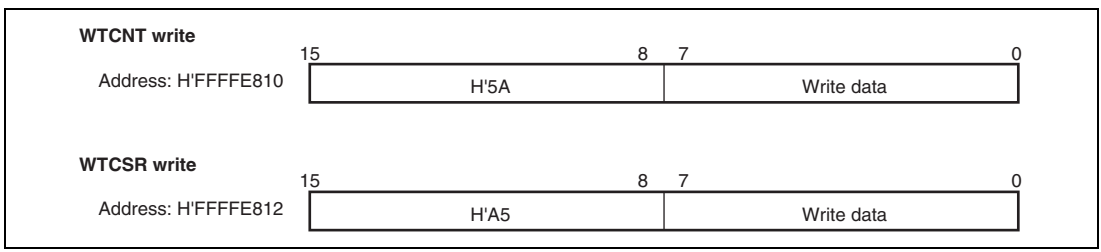



**Figure 13.2 Writing to WTCNT and WTCSR**

## 13.4 Operation

### 13.4.1 Revoking Software Standbys

The WDT can be used to revoke software standby mode with an NMI interrupt or external interrupt (IRQ). The procedure is described below. (The WDT does not run when resets are used for revoking, so keep the $\overline{\text{RES}}$ pin low until the clock stabilizes.)

1. Before 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 CKS2 to CKS0 bits in WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. Transition to software standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting a change in the level input to the NMI or IRQ pin.
5. When the WDT count overflows, the CPG starts supplying the clock and the LSI resumes operation. The WOVF flag in WTCSR is not set when this happens.

### 13.4.2 Using Watchdog Timer Mode

While operating in watchdog timer mode, the WDT generates an internal reset of the type specified by the RSTS bit in WTCSR and asserts a signal through the $\overline{\text{WDTOVF}}$ pin every time the counter overflows.

1. Set the WT/IT bit in WTCSR to 1, set the reset type in the RSTS bit, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the WTCNT counter.
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 prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1, asserts a signal through the $\overline{\text{WDTOVF}}$ pin for one cycle of the count clock specified by the CKS2 to CKS0 bits, and generates a reset of the type specified by the RSTS bit. The counter then resumes counting.

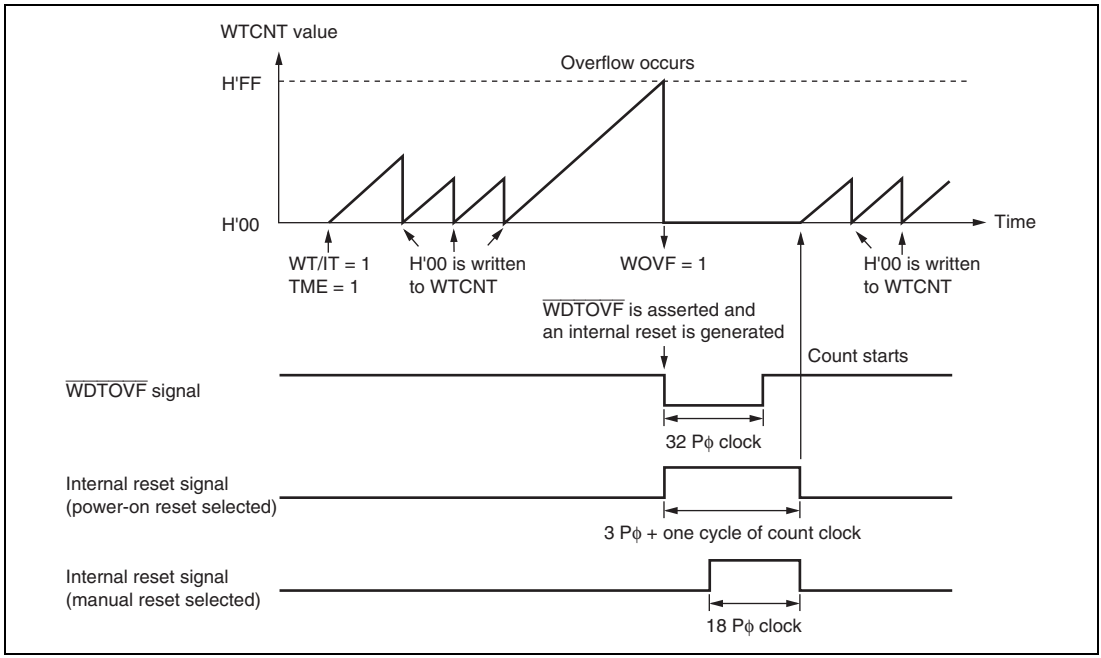



**Figure 13.3 Operation in Watchdog Timer Mode**
**(When WTCNT Count Clock is Specified to Pϕ/32 by CKS2 to CKS0)**

### 13.4.3 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/IT bit in WTCSR to 0, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the WTCNT counter.
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 flag in WTCSR to 1 and an interval timer interrupt request is sent to the INTC. The counter then resumes counting.

## 13.5 Interrupt Source

The WDT has one interrupt source: the interval timer interrupt (ITI).

Table 13.3 shows this interrupt source. An interval timer interrupt (ITI) is generated when the interval timer overflow flag bit (IOVF) in the watchdog timer control status register (WTSCR) is set to 1.

The interrupt request is canceled by clearing the interrupt flag to 0.

**Table 13.3 Interrupt Source**

| Name | Interrupt Source | Interrupt Enable Bit | Interrupt Flag Bit |
|---|---|---|---|
| ITI | Interval timer interrupt | — | Interval timer overflow flag (IOVF) |

## 13.6 Usage Note

### 13.6.1 WTCNT Setting Value

If WTCNT is set to H'FF in interval timer mode, overflow does not occur when WTCNT changes from H'FF to H'00 after one cycle of count clock, but overflow occurs when WTCNT changes from H'FF to H'00 after 257 cycles of count clock.

If WTCNT is set to H'FF in watchdog timer mode, overflow occurs when WTCNT changes from H'FF to H'00 after one cycle of count clock.

# Section 14 Serial Communication Interface (SCI)

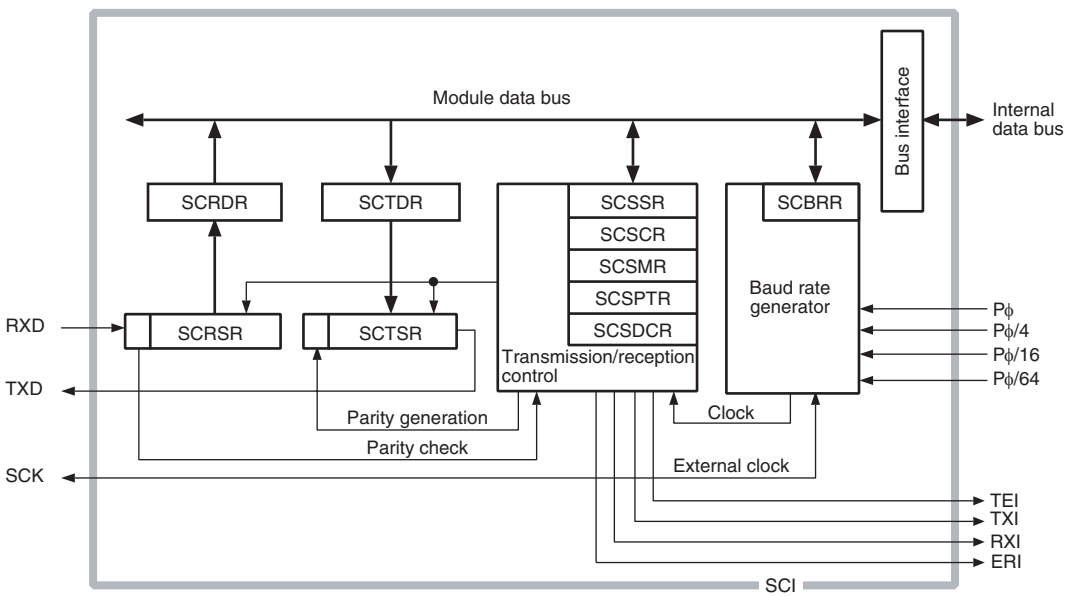
This LSI has three independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clock synchronous serial communication. In asynchronous serial communication mode, serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function).

## 14.1 Features

- Choice of asynchronous or clock synchronous serial communication mode
- Asynchronous mode:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are twelve selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Multiprocessor communications
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: Break is detected by reading the RXD pin level directly when a framing error occurs.
- Clock synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a clock synchronous communication function.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal clock) or SCK pin (external clock)
- Choice of LSB-first or MSB-first data transfer (except for 7-bit data in asynchronous mode)

- Four types of interrupts: There are four interrupt sources, transmit-data-empty, transmit end, receive-data-full, and receive error interrupts, and each interrupt can be requested independently. The data transfer controller (DTC) can be activated by the transmit-data-empty interrupt or receive-data-full interrupt to transfer data.
- Module standby mode can be set

Figure 14.1 shows a block diagram of the SCI.




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

**Figure 14.1 Block Diagram of SCI**

## 14.2 Input/Output Pins

The SCI has the serial pins summarized in table 14.1.

**Table 14.1 Pin Configuration**

| Channel | Pin Name* | I/O | Function |
|---|---|---|---|
| 0 | SCK0 | I/O | SCI0 clock input/output |
| | RXD0 | Input | SCI0 receive data input |
| | TXD0 | Output | SCI0 transmit data output |
| 1 | SCK1 | I/O | SCI1 clock input/output |
| | RXD1 | Input | SCI1 receive data input |
| | TXD1 | Output | SCI1 transmit data output |
| 2 | SCK2 | I/O | SCI2 clock input/output |
| | RXD2 | Input | SCI2 receive data input |
| | TXD2 | Output | SCI2 transmit data output |

Note: * Pin names SCK, RXD, and TXD are used in the description for all channels, omitting the channel designation.

## 14.3 Register Descriptions

The SCI has the following registers for each channel. For details on register addresses and register states during each processing, refer to section 25, List of Registers.

**Table 14.2 Register Configuration**

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 0 | Serial mode register_0 | SCSMR_0 | R/W | H'00 | H'FFFFC000 | 8 |
| | Bit rate register_0 | SCBRR_0 | R/W | H'FF | H'FFFFC002 | 8 |
| | Serial control register_0 | SCSCR_0 | R/W | H'00 | H'FFFFC004 | 8 |
| | Transmit data register_0 | SCTDR_0 | — | — | H'FFFFC006 | 8 |
| | Serial status register_0 | SCSSR_0 | R/W | H'84 | H'FFFFC008 | 8 |
| | Receive data register_0 | SCRDR_0 | — | — | H'FFFFC00A | 8 |
| | Serial direction control register_0 | SCSDCR_0 | R/W | H'F2 | H'FFFFC00C | 8 |
| | Serial port register_0 | SCSPTR_0 | R/W | H'0x | H'FFFFC00E | 8 |
| 1 | Serial mode register_1 | SCSMR_1 | R/W | H'00 | H'FFFFC080 | 8 |
| | Bit rate register_1 | SCBRR_1 | R/W | H'FF | H'FFFFC082 | 8 |
| | Serial control register_1 | SCSCR_1 | R/W | H'00 | H'FFFFC084 | 8 |
| | Transmit data register_1 | SCTDR_1 | — | — | H'FFFFC086 | 8 |
| | Serial status register_1 | SCSSR_1 | R/W | H'84 | H'FFFFC088 | 8 |
| | Receive data register_1 | SCRDR_1 | — | — | H'FFFFC08A | 8 |
| | Serial direction control register_1 | SCSDCR_1 | R/W | H'F2 | H'FFFFC08C | 8 |
| | Serial port register_1 | SCSPTR_1 | R/W | H'0x | H'FFFFC08E | 8 |
| 2 | Serial mode register_2 | SCSMR_2 | R/W | H'00 | H'FFFFC100 | 8 |
| | Bit rate register_2 | SCBRR_2 | R/W | H'FF | H'FFFFC102 | 8 |
| | Serial control register_2 | SCSCR_2 | R/W | H'00 | H'FFFFC104 | 8 |
| | Transmit data register_2 | SCTDR_2 | — | — | H'FFFFC106 | 8 |
| | Serial status register_2 | SCSSR_2 | R/W | H'84 | H'FFFFC108 | 8 |
| | Receive data register_2 | SCRDR_2 | — | — | H'FFFFC10A | 8 |
| | Serial direction control register_2 | SCSDCR_2 | R/W | H'F2 | H'FFFFC10C | 8 |
| | Serial port register_2 | SCSPTR_2 | R/W | H'0x | H'FFFFC10E | 8 |

### 14.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RXD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to SCRDR. The CPU cannot read or write to SCRSR directly.

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

### 14.3.2 Receive Data Register (SCRDR)

SCRDR is a register that stores serial receive data. After receiving one byte of serial data, the SCI transfers the received data from the receive shift register (SCRSR) into SCRDR for storage and completes operation. After that, SCRSR is ready to receive data.

Since SCRSR and SCRDR work as a double buffer in this way, data can be received continuously.

SCRDR is a read-only register and cannot be written to by the CPU.

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

### 14.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. The SCI loads transmit data from the transmit data register (SCTDR) into SCTSR, then transmits the data serially from the TXD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from SCTDR into SCTSR and starts transmitting again. If the TDRE flag in the serial status register (SCSSR) is set to 1, the SCI does not transfer data from SCTDR to SCTSR. The CPU cannot read or write to SCTSR directly.

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

### 14.3.4 Transmit Data Register (SCTDR)

SCTDR is an 8-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCTDR into SCTSR and starts serial transmission. If the next transmit data has been written to SCTDR during serial transmission from SCTSR, the SCI can transmit data continuously. SCTDR can always be written or read to by the CPU.

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

### 14.3.5 Serial Mode Register (SCSMR)

SCSMR is an 8-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | C/$\overline{A}$ | CHR | PE | O/$\overline{E}$ | STOP | MP | CKS[1:0] | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 | C/$\overline{A}$ | 0 | R/W | Communication Mode<br>Selects whether the SCI operates in asynchronous or clock synchronous mode.<br>0: Asynchronous mode<br>1: Clock synchronous mode |
| 6 | CHR | 0 | R/W | Character Length<br>Selects 7-bit or 8-bit data in asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR setting. When 7-bit data is selected, the MSB (bit 7) of the transmit data register is not transmitted.<br>0: 8-bit data<br>1: 7-bit data |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 5 | PE | 0 | R/W | Parity Enable<br>Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting.<br>0: Parity bit not added or checked<br>1: Parity bit added and checked*<br>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<br>Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is ignored in clock synchronous mode, or in asynchronous mode when parity addition and checking is disabled.<br>0: Even parity<br>1: Odd parity<br>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.<br>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<br>Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clock synchronous mode because no stop bits are added.<br>0: One stop bit*[1]<br>1: Two stop bits*[2]<br>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.<br>Notes: 1. When transmitting, a single 1-bit is added at the end of each transmitted character.<br>2. When transmitting, two 1 bits are added at the end of each transmitted character. |
| 2 | MP | 0 | R/W | Multiprocessor Mode (only in asynchronous mode)<br>Enables or disables multiprocessor mode. The PE and O/$\overline{E}$ bit settings are ignored in multiprocessor mode.<br>0: Multiprocessor mode disabled<br>1: Multiprocessor mode enabled |
| 1, 0 | CKS[1:0] | 00 | R/W | Clock Select 1 and 0<br>Select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 14.3.10, Bit Rate Register (SCBRR).<br>00: Pϕ<br>01: Pϕ/4<br>10: Pϕ/16<br>11: Pϕ/64<br>Note: Pϕ: Peripheral clock |

### 14.3.6 Serial Control Register (SCSCR)

SCSCR is an 8-bit register that enables or disables SCI transmission/reception and interrupt requests and selects the transmit/receive clock source. The CPU can always read and write to SCSCR.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE[1:0] | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 | TIE | 0 | R/W | Transmit Interrupt Enable<br>Enables or disables a transmit-data-empty interrupt (TXI) to be issued when the TDRE flag in the serial status register (SCSSR) is set to 1 after serial transmit data is sent from the transmit data register (SCTDR) to the transmit shift register (SCTSR).<br>TXI can be canceled by clearing the TDRE flag to 0 after reading TDRE = 1 or by clearing the TIE bit to 0.<br>0: Transmit-data-empty interrupt request (TXI) is disabled<br>1: Transmit-data-empty interrupt request (TXI) is enabled |
| 6 | RIE | 0 | R/W | Receive Interrupt Enable<br>Enables or disables a receive-data-full interrupt (RXI) and a receive error interrupt (ERI) to be issued when the RDRF flag in SCSSR is set to 1 after the serial data received is transferred from the receive shift register (SCRSR) to the receive data register (SCRDR).<br>RXI can be canceled by clearing the RDRF flag after reading RDRF =1. ERI can be canceled by clearing the FER, PER, or ORER flag to 0 after reading 1 from the flag. Both RXI and ERI can also be canceled by clearing the RIE bit to 0.<br>0: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled<br>1: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 5 | TE | 0 | R/W | Transmit Enable<br>Enables or disables the SCI serial transmitter.<br>0: Transmitter disabled*[1]<br>1: Transmitter enabled*[2]<br>Notes: 1. The TDRE flag in SCSSR is fixed at 1.<br>2. Serial transmission starts after writing transmit data into SCTDR and clearing the TDRE flag in SCSSR to 0 while the transmitter is enabled. Select the transmit format in the serial mode register (SCSMR) before setting TE to 1. |
| 4 | RE | 0 | R/W | Receive Enable<br>Enables or disables the SCI serial receiver.<br>0: Receiver disabled*[1]<br>1: Receiver enabled*[2]<br>Notes: 1. Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, and ORER). These flags retain their previous values.<br>2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in clock synchronous mode. Select the receive format in SCSMR before setting RE to 1. |
| 3 | MPIE | 0 | R/W | Multiprocessor Interrupt Enable (only when MP = 1 in SCSMR in asynchronous mode)<br>When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped and setting of the RDRF, FER, and ORER status flags in SCSSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared to 0 and normal receiving operation is resumed. For details, refer to section 14.4.4, Multiprocessor Communication Function. |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 2 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br>Enables or disables a transmit end interrupt (TEI) to be issued when no valid transmit data is found in SCTDR during MSB data transmission.<br>TEI can be canceled by clearing the TEND flag to 0 (by clearing the TDRE flag in SCSSR to 0 after reading TDRE = 1) or by clearing the TEIE bit to 0.<br>0: Transmit end interrupt request (TEI) is disabled<br>1: Transmit end interrupt request (TEI) is enabled |
| 1, 0 | CKE[1:0] | 00 | R/W | Clock Enable 1 and 0<br>Select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input.<br>When selecting the clock output in clock synchronous mode, set the C/$\overline{A}$ bit in SCSMR to 1 and then set bits CKE1 and CKE0. For details on clock source selection, refer to table 14.14 in section 14.4, Operation.<br>• Asynchronous mode<br>00: Internal clock, SCK pin used for input pin (The input signal is ignored.)<br>01: Internal clock, SCK pin used for clock output*[1]<br>10: External clock, SCK pin used for clock input*[2]<br>11: External clock, SCK pin used for clock input*[2]<br>• Clock synchronous mode<br>00: Internal clock, SCK pin used for synchronous clock output<br>01: Internal clock, SCK pin used for synchronous clock output<br>10: External clock, SCK pin used for synchronous clock input<br>11: External clock, SCK pin used for synchronous clock input<br>Notes: 1. The output clock frequency is 16 times the bit rate.<br>2. The input clock frequency is 16 times the bit rate. |

### 14.3.7 Serial Status Register (SCSSR)

SCSSR is an 8-bit register that contains status flags to indicate the SCI operating state.

The CPU can always read and write to SCSSR, but cannot write 1 to status flags TDRE, RDRF, ORER, PER, and FER. These flags can be cleared to 0 only after 1 is read from the flags. The TEND flag is a read-only bit and cannot be modified.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TDRE | RDRF | ORER | FER | PER | TEND | MPB | MPBT |
| Initial value: | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W: | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R/(W)* | R | R | R/W |

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

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 | TDRE | 1 | R/(W)* | Transmit Data Register Empty<br>Indicates whether data has been transferred from the transmit data register (SCTDR) to the transmit shift register (SCTSR) and SCTDR has become ready to be written with next serial transmit data.<br>0: Indicates that SCTDR holds valid transmit data<br>[Clearing conditions]<br>• When 0 is written to TDRE after reading TDRE = 1<br>• When the DTC is activated by a TXI interrupt and transmit data is transferred to SCTDR while the DISEL bit of MRB in the DTC is 0<br>1: Indicates that SCTDR does not hold valid transmit data<br>[Setting conditions]<br>• By a power-on reset or in standby mode<br>• When the TE bit in SCSCR is 0<br>• When data is transferred from SCTDR to SCTSR and data can be written to SCTDR |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 6 | RDRF | 0 | R/(W)* | Receive Data Register Full<br>Indicates that the received data is stored in the receive data register (SCRDR).<br>0: Indicates that valid received data is not stored in SCRDR<br>[Clearing conditions]<br>• By a power-on reset or in standby mode<br>• When 0 is written to RDRF after reading RDRF = 1<br>• When the DTC is activated by an RXI interrupt and data is transferred from SCRDR while the DISEL bit of MRB in the DTC is 0<br>1: Indicates that valid received data is stored in SCRDR<br>[Setting condition]<br>• When serial reception ends normally and receive data is transferred from SCRSR to SCRDR<br>Note: SCRDR and the RDRF flag are not affected and retain their previous states even if an error is detected during data reception or if the RE bit in the serial control register (SCSCR) is cleared to 0. If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the received data will be lost. |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 5 | ORER | 0 | R/(W)* | Overrun Error<br>Indicates that an overrun error occurred during reception, causing abnormal termination.<br>0: Indicates that reception is in progress or was completed successfully*[1]<br>[Clearing conditions]<br>• By a power-on reset or in standby mode<br>• When 0 is written to ORER after reading ORER = 1<br>1: Indicates that an overrun error occurred during reception*[2]<br>[Setting condition]<br>• When the next serial reception is completed while RDRF = 1<br>Notes: 1. The ORER flag is not affected and retains its previous value when the RE bit in SCSCR is cleared to 0.<br>2. The receive data prior to the overrun error is retained in SCRDR, and the data received subsequently is lost. Subsequent serial reception cannot be continued while the ORER flag is set to 1. |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 4 | FER | 0 | R/(W)* | Framing Error<br>Indicates that a framing error occurred during data reception in asynchronous mode, causing abnormal termination.<br>0: Indicates that reception is in progress or was completed successfully*[1]<br>[Clearing conditions]<br>• By a power-on reset or in standby mode<br>• When 0 is written to FER after reading FER = 1<br>1: Indicates that a framing error occurred during reception<br>[Setting condition]<br>• When the SCI founds that the stop bit at the end of the received data is 0 after completing reception*[2]<br>Notes: 1. The FER flag is not affected and retains its previous value when the RE bit in SCSCR is cleared to 0.<br>2. In 2-stop-bit mode, only the first stop bit is checked for a value to 1; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to SCRDR but the RDRF flag is not set. Subsequent serial reception cannot be continued while the FER flag is set to 1. |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 3 | PER | 0 | R/(W)* | Parity Error<br>Indicates that a parity error occurred during data reception in asynchronous mode, causing abnormal termination.<br>0: Indicates that reception is in progress or was completed successfully*[1]<br>[Clearing conditions]<br>• By a power-on reset or in standby mode<br>• When 0 is written to PER after reading PER = 1<br>1: Indicates that a parity error occurred during reception*[2]<br>[Setting condition]<br>• When the number of 1s in the received data and parity does not match the even or odd parity specified by the O/$\overline{E}$ bit in the serial mode register (SCSMR).<br>Notes: 1. The PER flag is not affected and retains its previous value when the RE bit in SCSCR is cleared to 0.<br>2. If a parity error occurs, the receive data is transferred to SCRDR but the RDRF flag is not set. Subsequent serial reception cannot be continued while the PER flag is set to 1. |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 2 | TEND | 1 | R | Transmit End<br>Indicates that no valid data was in SCTDR during transmission of the last bit of the transmit character and transmission has ended.<br>The TEND flag is read-only and cannot be modified.<br>0: Indicates that transmission is in progress<br>[Clearing condition]<br>• When 0 is written to TDRE after reading TDRE = 1<br>1: Indicates that transmission has ended<br>[Setting conditions]<br>• By a power-on reset or in standby mode<br>• When the TE bit in SCSCR is 0<br>• When TDRE = 1 during transmission of the last bit of a 1-byte serial transmit character<br>Note: The TEND flag value becomes undefined if data is written to SCTDR by activating the DTC by a TXI interrupt. In this case, do not use the TEND flag as the transmit end flag. |
| 1 | MPB | 0 | R | Multiprocessor Bit<br>Stores the multiprocessor bit found in the receive data. When the RE bit in SCSCR is cleared to 0, its previous state is retained. |
| 0 | MPBT | 0 | R/W | Multiprocessor Bit Transfer<br>Specifies the multiprocessor bit value to be added to the transmit frame. |

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

### 14.3.8 Serial Port Register (SCSPTR)

SCSPTR is an 8-bit register that controls input/output and data for the ports multiplexed with the SCI function pins. Data to be output through the TXD pin can be specified to control break of serial transfer. Through bits 3 and 2, data reading and writing through the SCK pin can be specified. Bit 7 enables or disables RXI interrupts. The CPU can always read and write to SCSPTR. When reading the value on the SCI pins, use the respective port register. For details, refer to section 21, I/O Ports.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EIO | - | - | - | SPB1IO | SPB1DT | SPB0IO | SPB0DT |
| Initial value: | 0 | 0 | 0 | 0 | 0 | - | 0 | 1 |
| R/W: | R/W | - | - | - | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 | EIO | 0 | R/W | Error Interrupt Only<br>Enables or disables RXI interrupts. While the EIO bit is set to 1, the SCI does not request an RXI interrupt to the CPU even if the RIE bit is set to 1.<br>0: The RIE bit enables or disables RXI and ERI interrupts. While the RIE bit is 1, RXI and ERI interrupts are sent to the INTC.<br>1: While the RIE bit is 1, only the ERI interrupt is sent to the INTC. |
| 6 to 4 | — | All 0 | — | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 3 | SPB1IO | 0 | R/W | Clock Port Input/Output in Serial Port<br>Specifies the input/output direction of the SCK pin in the serial port. To output the data specified in the SPB1DT bit through the SCK pin as a port output pin, set the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0.<br>0: Does not output the SPB1DT bit value through the SCK pin.<br>1: Outputs the SPB1DT bit value through the SCK pin. |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 2 | SPB1DT | Undefined | R/W | Clock Port Data in Serial Port<br>Specifies the data output through the SCK pin in the serial port. Output should be enabled by the SPB1IO bit (for details, refer to the SPB1IO bit description). When output is enabled, the SPB1DT bit value is output through the SCK pin.<br>0: Low level is output<br>1: High level is output |
| 1 | SPB0IO | 0 | R/W | Serial Port Break Output<br>Together with the SPB0DT bit and the TE bit in SCSCR, controls the TXD pin. |
| 0 | SPB0DT | 1 | R/W | Serial Port Break Data<br>Together with the SPB0IO bit and TE bit in SCSCR, controls the TXD pin. Note that the TXD pin function needs to have been selected with the pin function controller (PFC). |

| TE bit setting in SCSCR | SPB0IO bit setting | SPB0DT bit setting | State of TXD pin |
|---|---|---|---|
| 0 | 0 | * | SPB0DT output disabled (initial state) |
| 0 | 1 | 0 | Output, low level |
| 0 | 1 | 1 | Output, high level |
| 1 | * | * | Output for transmit data in accord with the serial core logic |

Note: * Don't care

### 14.3.9 Serial Direction Control Register (SCSDCR)

The DIR bit in the serial direction control register (SCSDCR) selects LSB-first or MSB-first transfer. With an 8-bit data length, LSB-first/MSB-first selection is available regardless of the communication mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | DIR | - | - | - |
| Initial value: | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| R/W: | R | R | R | R | R/W | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 4 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. |
| 3 | DIR | 0 | R/W | Data Transfer Direction<br>Selects the serial/parallel conversion format. Valid for an 8-bit transmit/receive format.<br>0: SCTDR contents are transmitted in LSB-first order<br>Receive data is stored in SCRDR in LSB-first<br>1: SCTDR contents are transmitted in MSB-first order<br>Receive data is stored in SCRDR in MSB-first |
| 2 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 1 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 0 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

### 14.3.10 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR.

The SCBRR setting is calculated as follows:

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

- Asynchronous mode:

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

- Clock synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting value should satisfy the electrical characteristics.)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 14.3.)

**Table 14.3 SCSMR Settings**

| n | Clock Source | SCSMR Settings CKS1 | CKS0 |
|---|---|---|---|
| 0 | Pϕ | 0 | 0 |
| 1 | Pϕ/4 | 0 | 1 |
| 2 | Pϕ/16 | 1 | 0 |
| 3 | Pϕ/64 | 1 | 1 |

Note: The bit rate error in asynchronous is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Tables 14.4 to 14.6 show examples of SCBRR settings in asynchronous mode, and tables 14.7 to 14.9 show examples of SCBRR settings in clock synchronous mode.

**Table 14.4 Bit Rates and SCBRR Settings in Asynchronous Mode (1)**

| Bit Rate (bits/s) | Pϕ (MHz) 10 | | | 12 | | | 14 | | | 16 | | | 18 | | | 20 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 177 | -0.25 | 2 | 212 | 0.03 | 2 | 248 | -0.17 | 3 | 70 | 0.03 | 3 | 79 | -0.12 | 3 | 88 | -0.25 |
| 150 | 2 | 129 | 0.16 | 2 | 155 | 0.16 | 2 | 181 | 0.16 | 2 | 207 | 0.16 | 2 | 233 | 0.16 | 3 | 64 | 0.16 |
| 300 | 2 | 64 | 0.16 | 2 | 77 | 0.16 | 2 | 90 | 0.16 | 2 | 103 | 0.16 | 2 | 116 | 0.16 | 2 | 129 | 0.16 |
| 600 | 1 | 129 | 0.16 | 1 | 155 | 0.16 | 1 | 181 | 0.16 | 1 | 207 | 0.16 | 1 | 233 | 0.16 | 2 | 64 | 0.16 |
| 1200 | 1 | 64 | 0.16 | 1 | 77 | 0.16 | 1 | 90 | 0.16 | 1 | 103 | 0.16 | 1 | 116 | 0.16 | 1 | 129 | 0.16 |
| 2400 | 0 | 129 | 0.16 | 0 | 155 | 0.16 | 0 | 181 | 0.16 | 0 | 207 | 0.16 | 0 | 233 | 0.16 | 1 | 64 | 0.16 |
| 4800 | 0 | 64 | 0.16 | 0 | 77 | 0.16 | 0 | 90 | 0.16 | 0 | 103 | 0.16 | 0 | 116 | 0.16 | 0 | 129 | 0.16 |
| 9600 | 0 | 32 | -1.36 | 0 | 38 | 0.16 | 0 | 45 | -0.93 | 0 | 51 | 0.16 | 0 | 58 | -0.69 | 0 | 64 | 0.16 |
| 14400 | 0 | 21 | -1.36 | 0 | 25 | 0.16 | 0 | 29 | 1.27 | 0 | 34 | -0.79 | 0 | 38 | 0.16 | 0 | 42 | 0.94 |
| 19200 | 0 | 15 | 1.73 | 0 | 19 | -2.34 | 0 | 22 | -0.93 | 0 | 25 | 0.16 | 0 | 28 | 1.02 | 0 | 32 | -1.36 |
| 28800 | 0 | 10 | -1.36 | 0 | 12 | 0.16 | 0 | 14 | 1.27 | 0 | 16 | 2.12 | 0 | 19 | -2.34 | 0 | 21 | -1.36 |
| 31250 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 13 | 0.00 | 0 | 15 | 0.00 | 0 | 17 | 0.00 | 0 | 19 | 0.00 |
| 38400 | 0 | 7 | 1.73 | 0 | 9 | -2.34 | 0 | 10 | 3.57 | 0 | 12 | 0.16 | 0 | 14 | -2.34 | 0 | 15 | 1.73 |

## Table 14.5 Bit Rates and SCBRR Settings in Asynchronous Mode (2)

| Bit Rate (bits/s) | Pϕ (MHz) 22 | | | 24 | | | 26 | | | 28 | | | 30 | | | 32 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 3 | 97 | -0.35 | 3 | 106 | -0.44 | 3 | 114 | 0.36 | 3 | 123 | 0.23 | 3 | 132 | 0.13 | 3 | 141 | 0.03 |
| 150 | 3 | 71 | -0.54 | 3 | 77 | 0.16 | 3 | 84 | -0.43 | 3 | 90 | 0.16 | 3 | 97 | -0.35 | 3 | 103 | 0.16 |
| 300 | 2 | 142 | 0.16 | 2 | 155 | 0.16 | 2 | 168 | 0.16 | 2 | 181 | 0.16 | 2 | 194 | 0.16 | 2 | 207 | 0.16 |
| 600 | 2 | 71 | -0.54 | 2 | 77 | 0.16 | 2 | 84 | -0.43 | 2 | 90 | 0.16 | 2 | 97 | -0.35 | 2 | 103 | 0.16 |
| 1200 | 1 | 142 | 0.16 | 1 | 155 | 0.16 | 1 | 168 | 0.16 | 1 | 181 | 0.16 | 1 | 194 | 0.16 | 1 | 207 | 0.16 |
| 2400 | 1 | 71 | -0.54 | 1 | 77 | 0.16 | 1 | 84 | -0.43 | 1 | 90 | 0.16 | 1 | 97 | -0.35 | 1 | 103 | 0.16 |
| 4800 | 0 | 142 | 0.16 | 0 | 155 | 0.16 | 0 | 168 | 0.16 | 0 | 181 | 0.16 | 0 | 194 | 0.16 | 0 | 207 | 0.16 |
| 9600 | 0 | 71 | -0.54 | 0 | 77 | 0.16 | 0 | 84 | -0.43 | 0 | 90 | 0.16 | 0 | 97 | -0.35 | 0 | 103 | 0.16 |
| 14400 | 0 | 47 | -0.54 | 0 | 51 | 0.16 | 0 | 55 | 0.76 | 0 | 60 | -0.39 | 0 | 64 | 0.16 | 0 | 68 | 0.64 |
| 19200 | 0 | 35 | -0.54 | 0 | 38 | 0.16 | 0 | 41 | 0.76 | 0 | 45 | -0.93 | 0 | 48 | -0.35 | 0 | 51 | 0.16 |
| 28800 | 0 | 23 | -0.54 | 0 | 25 | 0.16 | 0 | 27 | 0.76 | 0 | 29 | 1.27 | 0 | 32 | -1.36 | 0 | 34 | -0.79 |
| 31250 | 0 | 21 | 0.00 | 0 | 23 | 0.00 | 0 | 25 | 0.00 | 0 | 27 | 0.00 | 0 | 29 | 0.00 | 0 | 31 | 0.00 |
| 38400 | 0 | 17 | -0.54 | 0 | 19 | -2.34 | 0 | 20 | 0.76 | 0 | 22 | -0.93 | 0 | 23 | 1.73 | 0 | 25 | 0.16 |

## Table 14.6 Bit Rates and SCBRR Settings in Asynchronous Mode (3)

| Bit Rate (bits/s) | Pϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 34 | | | 36 | | | 38 | | | 40 | | |
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 3 | 150 | -0.05 | 3 | 159 | -0.12 | 3 | 168 | -0.19 | 3 | 177 | -0.25 |
| 150 | 3 | 110 | -0.29 | 3 | 116 | 0.16 | 3 | 123 | -0.24 | 3 | 129 | 0.16 |
| 300 | 2 | 220 | 0.16 | 2 | 233 | 0.16 | 2 | 246 | 0.16 | 3 | 64 | 0.16 |
| 600 | 2 | 110 | -0.29 | 2 | 116 | 0.16 | 2 | 123 | -0.24 | 2 | 129 | 0.16 |
| 1200 | 1 | 220 | 0.16 | 1 | 233 | 0.16 | 1 | 246 | 0.16 | 2 | 64 | 0.16 |
| 2400 | 1 | 110 | -0.29 | 1 | 116 | 0.16 | 1 | 123 | -0.24 | 1 | 129 | 0.16 |
| 4800 | 0 | 220 | 0.16 | 0 | 233 | 0.16 | 0 | 246 | 0.16 | 1 | 64 | 0.16 |
| 9600 | 0 | 110 | -0.29 | 0 | 116 | 0.16 | 0 | 123 | -0.24 | 0 | 129 | 0.16 |
| 14400 | 0 | 73 | -0.29 | 0 | 77 | 0.16 | 0 | 81 | 0.57 | 0 | 86 | -0.22 |
| 19200 | 0 | 54 | 0.62 | 0 | 58 | -0.69 | 0 | 61 | -0.24 | 0 | 64 | 0.16 |
| 28800 | 0 | 36 | -0.29 | 0 | 38 | 0.16 | 0 | 40 | 0.57 | 0 | 42 | 0.94 |
| 31250 | 0 | 33 | 0.00 | 0 | 35 | 0.00 | 0 | 37 | 0.00 | 0 | 39 | 0.00 |
| 38400 | 0 | 27 | -1.18 | 0 | 28 | 1.02 | 0 | 30 | -0.24 | 0 | 32 | -1.36 |

**Table 14.7 Bit Rates and SCBRR Settings in Clock Synchronous Mode (1)**

| Bit Rate (bits/s) | Pϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | | 12 | | 14 | | 16 | | 18 | | 20 | |
| | n | N | n | N | n | N | n | N | n | N | n | N |
| 250 | 3 | 155 | 3 | 187 | 3 | 218 | 3 | 249 | | | | |
| 500 | 3 | 77 | 3 | 93 | 3 | 108 | 3 | 124 | 3 | 140 | 3 | 155 |
| 1000 | 2 | 155 | 2 | 187 | 2 | 218 | 2 | 249 | 3 | 69 | 3 | 77 |
| 2500 | 1 | 249 | 2 | 74 | 2 | 87 | 2 | 99 | 2 | 112 | 2 | 124 |
| 5000 | 1 | 124 | 1 | 149 | 1 | 174 | 1 | 199 | 1 | 224 | 1 | 249 |
| 10000 | 0 | 249 | 1 | 74 | 1 | 87 | 1 | 99 | 1 | 112 | 1 | 124 |
| 25000 | 0 | 99 | 0 | 119 | 0 | 139 | 0 | 159 | 0 | 179 | 0 | 199 |
| 50000 | 0 | 49 | 0 | 59 | 0 | 69 | 0 | 79 | 0 | 89 | 0 | 99 |
| 100000 | 0 | 24 | 0 | 29 | 0 | 34 | 0 | 39 | 0 | 44 | 0 | 49 |
| 250000 | 0 | 9 | 0 | 11 | 0 | 13 | 0 | 15 | 0 | 17 | 0 | 19 |
| 500000 | 0 | 4 | 0 | 5 | 0 | 6 | 0 | 7 | 0 | 8 | 0 | 9 |
| 1000000 | — | — | 0 | 2 | — | — | 0 | 3 | — | — | 0 | 4 |
| 2500000 | 0 | 0* | — | — | — | — | — | — | — | — | 0 | 1 |
| 5000000 | | | — | — | — | — | — | — | — | — | 0 | 0* |

**Table 14.8 Bit Rates and SCBRR Settings in Clock Synchronous Mode (2)**

| Bit Rate (bits/s) | Pϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 22 | | 24 | | 26 | | 28 | | 30 | | 32 | |
| | n | N | n | N | n | N | n | N | n | N | n | N |
| 250 | | | | | | | | | | | | |
| 500 | 3 | 171 | 3 | 187 | 3 | 202 | 3 | 218 | 3 | 233 | 3 | 249 |
| 1000 | 3 | 85 | 3 | 93 | 3 | 101 | 3 | 108 | 3 | 116 | 3 | 124 |
| 2500 | 2 | 137 | 2 | 149 | 2 | 162 | 2 | 174 | 2 | 187 | 2 | 199 |
| 5000 | 2 | 68 | 2 | 74 | 2 | 80 | 2 | 87 | 2 | 93 | 2 | 99 |
| 10000 | 1 | 137 | 1 | 149 | 1 | 162 | 1 | 174 | 1 | 187 | 1 | 199 |
| 25000 | 0 | 219 | 0 | 239 | 1 | 64 | 1 | 69 | 1 | 74 | 1 | 79 |
| 50000 | 0 | 109 | 0 | 119 | 0 | 129 | 0 | 139 | 0 | 149 | 0 | 159 |
| 100000 | 0 | 54 | 0 | 59 | 0 | 64 | 0 | 69 | 0 | 74 | 0 | 79 |
| 250000 | 0 | 21 | 0 | 23 | 0 | 25 | 0 | 27 | 0 | 29 | 0 | 31 |
| 500000 | 0 | 10 | 0 | 11 | 0 | 12 | 0 | 13 | 0 | 14 | 0 | 15 |
| 1000000 | — | — | 0 | 5 | — | — | 0 | 6 | — | — | 0 | 7 |
| 2500000 | — | — | — | — | — | — | — | — | 0 | 2 | — | — |
| 5000000 | — | — | — | — | — | — | — | — | — | — | — | — |

**Table 14.9 Bit Rates and SCBRR Settings in Clock Synchronous Mode (3)**

| Bit Rate (bits/s) | Pϕ (MHz) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 34 | | 36 | | 38 | | 40 | |
| | n | N | n | N | n | N | n | N |
| 250 | | | | | | | | |
| 500 | | | | | | | | |
| 1000 | 3 | 132 | 3 | 140 | 3 | 147 | 3 | 155 |
| 2500 | 2 | 212 | 2 | 224 | 2 | 237 | 2 | 249 |
| 5000 | 2 | 105 | 2 | 112 | 2 | 118 | 2 | 124 |
| 10000 | 1 | 212 | 1 | 224 | 1 | 237 | 1 | 249 |
| 25000 | 1 | 84 | 1 | 89 | 1 | 94 | 1 | 99 |
| 50000 | 0 | 169 | 0 | 179 | 0 | 189 | 0 | 199 |
| 100000 | 0 | 84 | 0 | 89 | 0 | 94 | 0 | 99 |
| 250000 | 0 | 33 | 0 | 35 | 0 | 37 | 0 | 39 |
| 500000 | 0 | 16 | 0 | 17 | 0 | 18 | 0 | 19 |
| 1000000 | — | — | 0 | 8 | — | — | 0 | 9 |
| 2500000 | — | — | — | — | — | — | 0 | 3 |
| 5000000 | — | — | — | — | — | — | 0 | 1 |

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

*: Continuous transmission/reception is disabled.

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

Table 14.10 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 14.11 and 14.12 list the maximum rates for external clock input.

**Table 14.10 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)**

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

**Table 14.11 Maximum Bit Rates with External Clock Input (Asynchronous Mode)**

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

**Table 14.12 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode)**

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

## 14.4 Operation

### 14.4.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses.

Asynchronous or clock synchronous mode is selected and the transmit format is specified in the serial mode register (SCSMR) as shown in table 14.13. The SCI clock source is selected by the combination of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in the serial control register (SCSCR) as shown in table 14.14.

Asynchronous Mode

- Data length is selectable: 7 or 8 bits.
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and breaks.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the clock supplied by the on-chip baud rate generator and can output a clock with a frequency 16 times the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Clock Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

**Table 14.13 SCSMR Settings and SCI Communication Formats**

| SCSMR Settings | | | | | SCI Communication Format | | |
|---|---|---|---|---|---|---|---|
| Bit 7 C/$\overline{A}$ | Bit 6 CHR | Bit 5 PE | Bit 3 STOP | Mode | Data Length | Parity Bit | Stop Bit Length |
| 0 | 0 | 0 | 0 | Asynchronous | 8-bit | Not set | 1 bit |
| | | | 1 | | | | 2 bits |
| | | 1 | 0 | | | Set | 1 bit |
| | | | 1 | | | | 2 bits |
| | 1 | 0 | 0 | | 7-bit | Not set | 1 bit |
| | | | 1 | | | | 2 bits |
| | | 1 | 0 | | | Set | 1 bit |
| | | | 1 | | | | 2 bits |
| 1 | x | x | x | Clock synchronous | 8-bit | Not set | None |

[Legend]

x: Don't care

**Table 14.14 SCSMR and SCSCR Settings and SCI Clock Source Selection**

| SCSMR | SCSCR Settings | | | | |
|---|---|---|---|---|---|
| Bit 7 C/$\overline{A}$ | Bit 1 CKE1 | Bit 0 CKE0 | Mode | Clock Source | SCK Pin Function |
| 0 | 0 | 0 | Asynchronous | Internal | SCI does not use the SCK pin. |
| | | 1 | | | Clock with a frequency 16 times the bit rate is output. |
| | 1 | 0 | | External | Input a clock with frequency 16 times the bit rate. |
| | | 1 | | | |
| 1 | 0 | 0 | Clock synchronous | Internal | Serial clock is output. |
| | | 1 | | | |
| | 1 | 0 | | External | Input the serial clock. |
| | | 1 | | | |

### 14.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 14.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

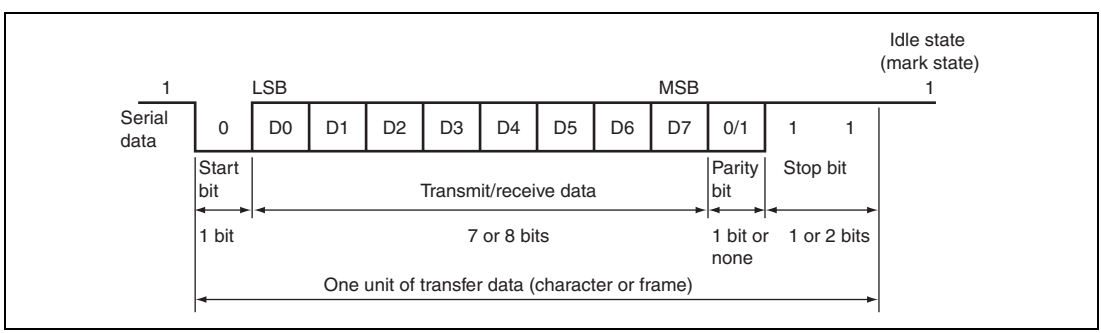



**Figure 14.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)**

### (1) Transmit/Receive Formats

Table 14.15 shows the transfer formats that can be selected in asynchronous mode. Any of 12 transfer formats can be selected according to the SCSMR settings.

**Table 14.15 Serial Transfer Formats (Asynchronous Mode)**

| SCSMR Settings | | | | Serial Transfer Format and Frame Length | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CHR | PE | MP | STOP | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 0 | 0 | 0 | 0 | S | 8-bit data | | | | | | | | STOP | | |
| 0 | 0 | 0 | 1 | S | 8-bit data | | | | | | | | STOP | STOP | |
| 0 | 1 | 0 | 0 | S | 8-bit data | | | | | | | | P | STOP | |
| 0 | 1 | 0 | 1 | S | 8-bit data | | | | | | | | P | STOP | STOP |
| 1 | 0 | 0 | 0 | S | 7-bit data | | | | | | | STOP | | | |
| 1 | 0 | 0 | 1 | S | 7-bit data | | | | | | | STOP | STOP | | |
| 1 | 1 | 0 | 0 | S | 7-bit data | | | | | | | P | STOP | | |
| 1 | 1 | 0 | 1 | S | 7-bit data | | | | | | | P | STOP | STOP | |
| 0 | x | 1 | 0 | S | 8-bit data | | | | | | | | MPB | STOP | |
| 0 | x | 1 | 1 | S | 8-bit data | | | | | | | | MPB | STOP | STOP |
| 1 | x | 1 | 0 | S | 7-bit data | | | | | | | MPB | STOP | | |
| 1 | x | 1 | 1 | S | 7-bit data | | | | | | | MPB | STOP | STOP | |

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit
x: Don't care

### (2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR) (table 14.14).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to 16 times the desired bit rate.

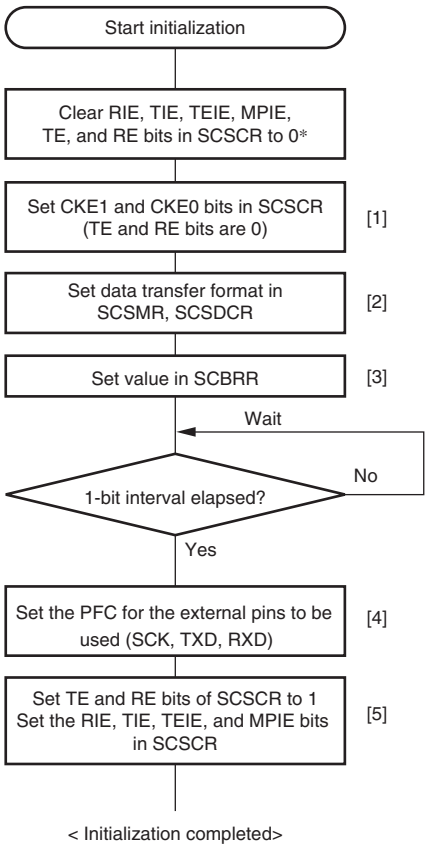
### (3) Transmitting and Receiving Data

**SCI Initialization (Asynchronous Mode):**

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI as follows.

When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing the TE bit to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR). Clearing the RE bit to 0, however, does not initialize the RDRF, PER, FER, and ORER flags or receive data register (SCRDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.




[1] Set the clock selection in SCSCR.

[2] Set the data transfer format in SCSMR and SCSDCR.

[3] Write a value corresponding to the bit rate to SCBRR. Not necessary if an external clock is used.

[4] Set PFC of the external pin used. Set RXD input during receiving and TXD output during transmitting. Set SCK input/output according to contents set by CKE1 and CKE0. When CKE1 and CKE0 are 0 in asynchronous mode, setting the SCK pin is unnecessary. Outputting clocks from the SCK pin starts at synchronous clock output setting.

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

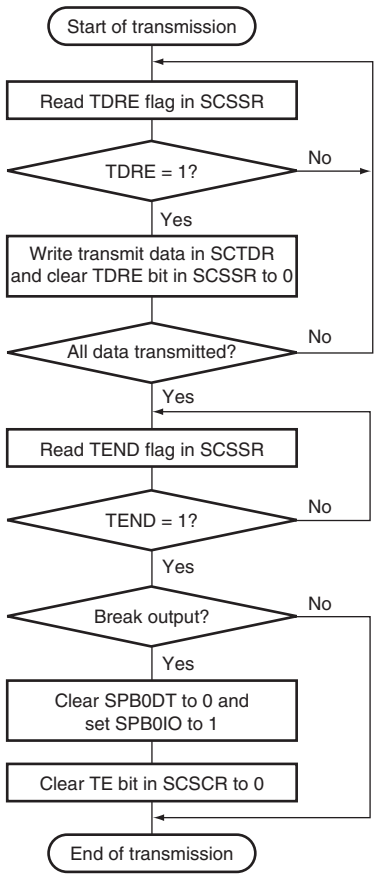
Note : * In simultaneous transmit/receive operation, the TE and RE bits must be cleared to 0 or set to 1 simultaneously.

**Figure 14.3 Sample Flowchart for SCI Initialization**

**Transmitting Serial Data (Asynchronous Mode):**

Figure 14.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCI for transmission.




[1] SCI status check and transmit data write:

Read SCSSR and check that the TDRE flag is set to 1, then write transmit data to SCTDR, and clear the TDRE flag to 0.

[2] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to SCTDR, and then clear the TDRE flag to 0.

Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to SCTDR.

[3] Break output at the end of serial transmission:

To output a break in serial transmission, clear the SPB0DT bit to 0 and set the SPB0IO bit to 1 in SCSPTR, then clear the TE bit in SCSCR to 0.

**Figure 14.4 Sample Flowchart for Transmitting Serial Data**

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in the serial status register (SCSSR). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR) and transfers the data from SCTDR to the transmit shift register (SCTSR).
2. After transferring data from SCTDR to SCTSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in the serial control register (SCSCR) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TXD pin in the following order.

   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   C. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multiprocessor bit is output. (A format in which neither parity nor multiprocessor bit is output can also be selected.)
   D. Stop bit(s): One or two 1 bits (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCI checks the TDRE flag at the timing for sending the stop bit.
   If the TDRE flag is 0, the data is transferred from SCTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.
   If the TDRE flag is 1, the TEND flag in SCSSR is set to 1, the stop bit is sent, and then the "mark state" is entered in which 1 is output. If the TEIE bit in SCSCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 14.5 shows an example of the operation for transmission.

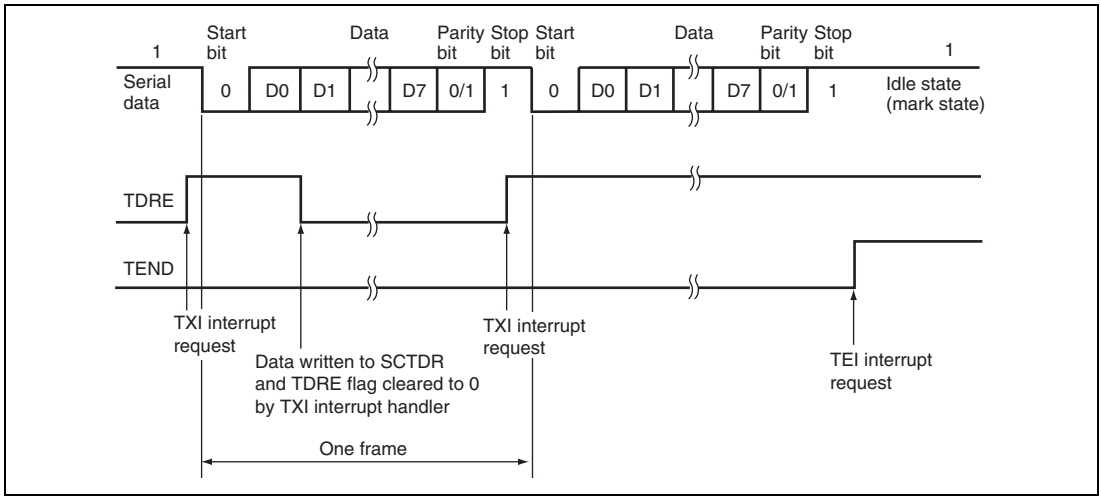
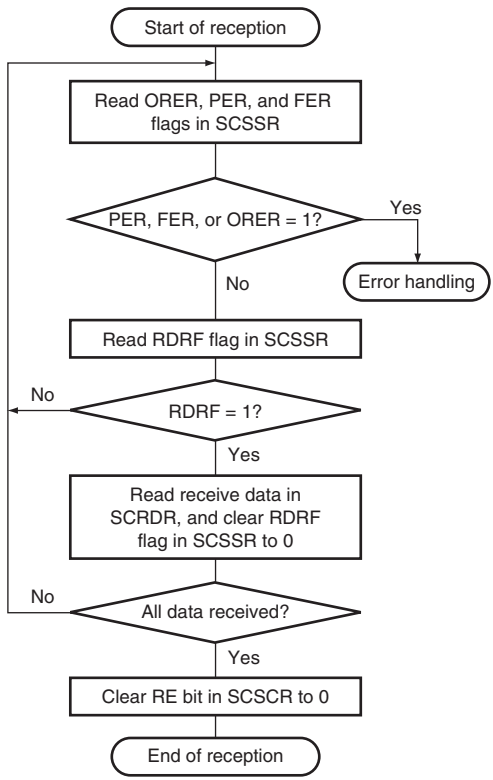



**Figure 14.5 Example of Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)**

**Receiving Serial Data (Asynchronous Mode):**

Figures 14.6 and 14.7 show a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCI for reception.




[1] Receive error handling and break detection:

If a receive error occurs, read the ORER, PER, and FER flags in SCSSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can also be detected by reading the value of the RXD pin.

[2] SCI status check and receive data read:

Read SCSSR and check that RDRF = 1, then read the receive data in SCRDR clear the RDRF flag to 0.

[3] Serial reception continuation procedure:

To continue serial reception, clear the RDRF flag to 0 before the stop bit for the current frame is received. The RDRF flag is cleared automatically when the data transfer controller (DTC) is activated to read the SCRDR value, and this step is not needed.

**Figure 14.6 Sample Flowchart for Receiving Serial Data (1)**

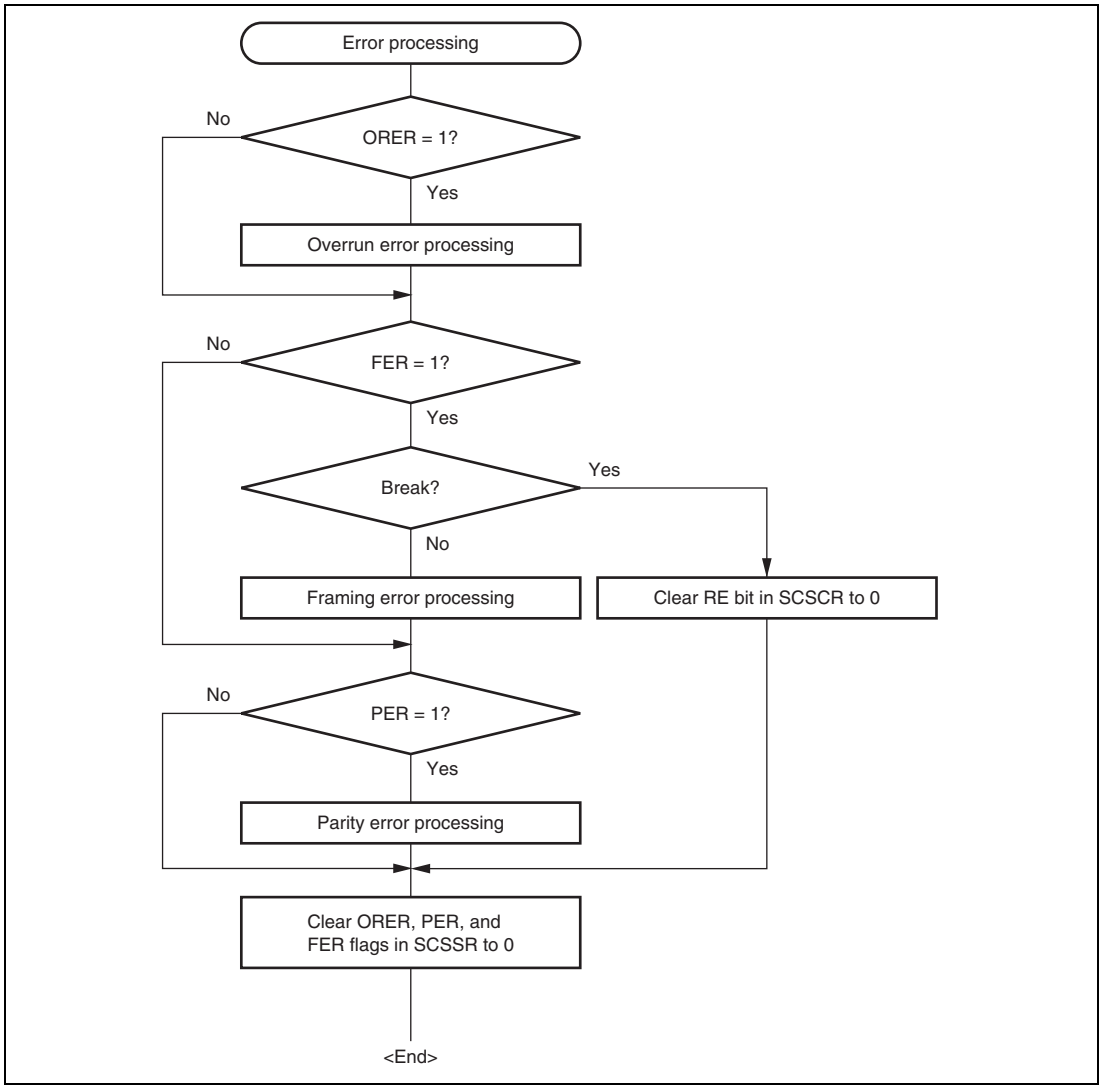



**Figure 14.7 Sample Flowchart for Receiving Serial Data (2)**

In serial reception, the SCI operates as described below.

1. The SCI monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.
   After receiving these bits, the SCI carries out the following checks.
   A. Parity check: The SCI counts the number of 1s in the received data and checks whether the count matches the even or odd parity specified by the O/$\overline{E}$ bit in the serial mode register (SCSMR).
   B. Stop bit check: The SCI checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   C. Status check: The SCI checks whether the RDRF flag is 0 and the received data can be transferred from the receive shift register (SCRSR) to SCRDR.

   If all the above checks are passed, the RDRF flag is set to 1 and the received data is stored in SCRDR. If a receive error is detected, the SCI operates as shown in table 14.16

   Note: When a receive error occurs, subsequent reception cannot be continued. In addition, the RDRF flag will not be set to 1 after reception; be sure to clear the error flag to 0.

4. If the EIO bit in SCSPTR is cleared to 0 and the RIE bit in SCSCR is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated. If the RIE bit in SCSCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive error interrupt (ERI) request is generated.

**Table 14.16 Receive Errors and Error Conditions**

| Receive Error | Abbreviation | Error Condition | Data Transfer |
|---|---|---|---|
| Overrun error | ORER | When the next data reception is completed while the RDRF flag in SCSSR is set to 1 | The received data is not transferred from SCRSR to SCRDR. |
| Framing error | FER | When the stop bit is 0 | The received data is transferred from SCRSR to SCRDR. |
| Parity error | PER | When the received data does not match the even or odd parity specified in SCSMR | The received data is transferred from SCRSR to SCRDR. |

Figure 14.8 shows an example of the operation for reception.

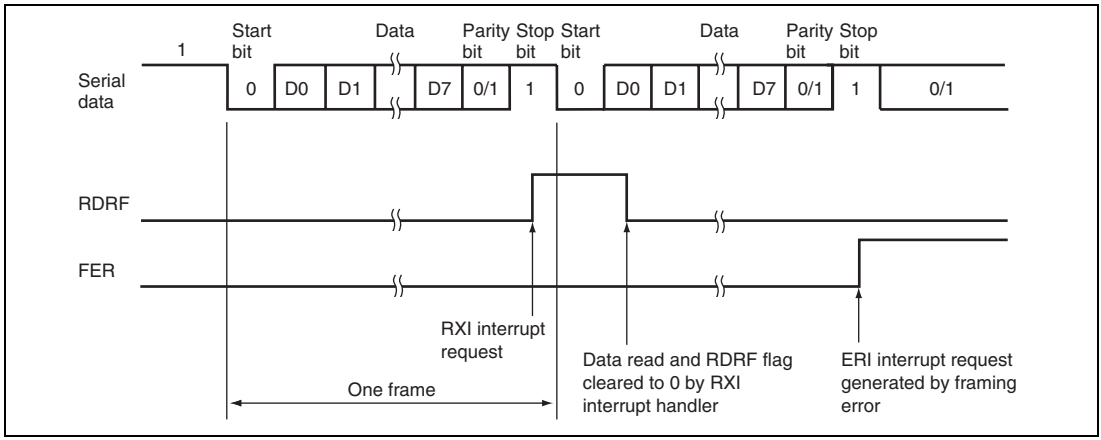



**Figure 14.8 Example of SCI Receive Operation (8-Bit Data, Parity, One Stop Bit)**

### 14.4.3 Clock Synchronous Mode

In clock synchronous mode, the SCIF transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 14.9 shows the general format in clock synchronous serial communication.

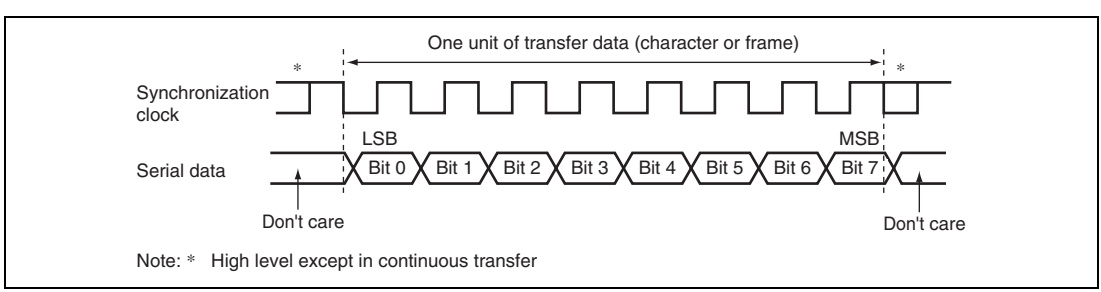



**Figure 14.9 Data Format in Clock Synchronous Communication**

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In clock synchronous mode, the SCI transmits or receives data by synchronizing with the rising edge of the serial clock.

### (1) Communication Format

The data length is fixed at eight bits. No parity bit can be added.

### (2) Clock

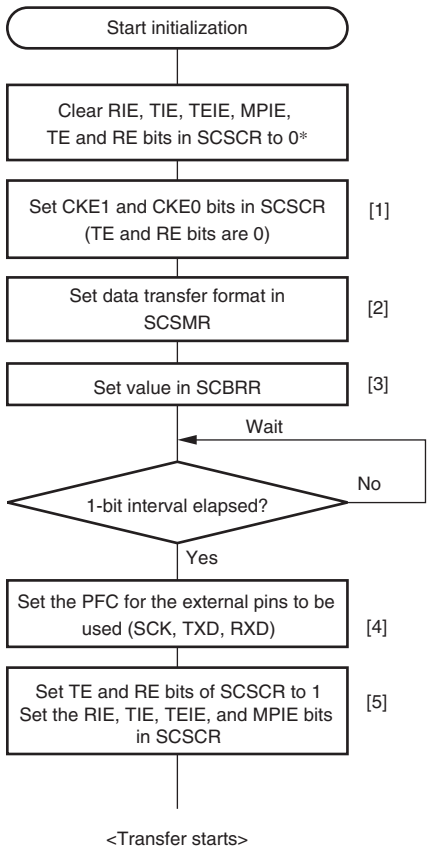
An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. For selection of the SCI clock source, see table 14.14.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. When only reception is performed, output of the synchronous clock continues until an overrun error occurs or the RE bit is cleared to 0. For the reception of n characters, select the external clock as the clock source. If the internal clock has to be used, set RE and TE to 1, then transmit n characters of dummy data at the same time as receiving the n characters of data.

### (3) Transmitting and Receiving Data

**SCI Initialization (Clock Synchronous Mode):** Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI. Clearing TE to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (SCRDR), which retain their previous contents.

Figure 14.10 shows a sample flowchart for initializing the SCI.




[1] Set the clock selection in SCSCR.

[2] Set the data transfer format in SCSMR.

[3] Write a value corresponding to the bit rate to SCBRR. Not necessary if an external clock is used.

[4] Set PFC of the external pin used. Set RXD input during receiving and TXD output during transmitting. Set SCK input/output according to contents set by CKE1 and CKE0.

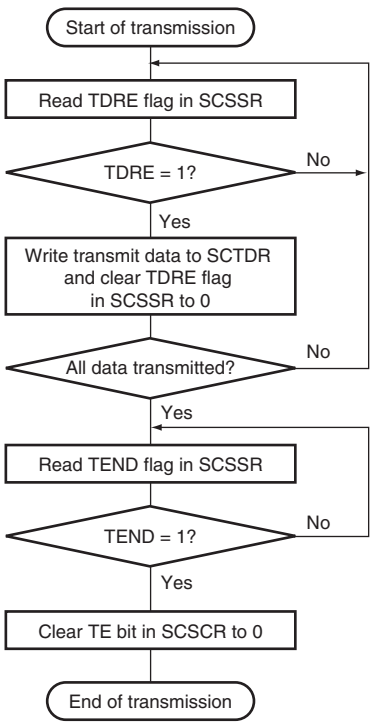
[5] Set the TE bit or RE bit in SCR to 1.* Also make settings of the RIE, TIE, TEIE, and MPIE bits. At this time, the TXD, RXD, and SCK pins are ready to be used. The TXD pin is in a mark state during transmitting. When synchronous clock output (clock master) is set during receiving in clock synchronous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

Note: * In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

**Figure 14.10 Sample Flowchart for SCI Initialization**

**Transmitting Serial Data (Clock Synchronous Mode):** Figure 14.11 shows a sample flowchart for transmitting serial data.

Use the following procedure for serial data transmission after enabling the SCI for transmission.




[1] SCI status check and transmit data write:

Read SCSSR and check that the TDRE flag is set to 1, then write transmit data to SCTDR, and clear the TDRE flag to 0.

[2] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to SCTDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to SCTDR.

**Figure 14.11 Sample Flowchart for Transmitting Serial Data**

**In transmitting serial data, the SCI operates as follows:**

1. The SCI monitors the TDRE flag in the serial status register (SCSSR). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR) and transfers the data from SCTDR to the transmit shift register (SCTSR).
2. After transferring data from SCTDR to SCTSR, the SCI sets the TDRE flag to 1 and starts transmission. If the transmit-data-empty interrupt enable bit (TIE) in the serial control register (SCSCR) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data is output from the TXD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7). If the TDRE flag is 0, the data is transferred from SCTDR to SCTSR and serial transmission of the next frame is started, If the TDRE flag is 1, the TEND flag in SCSSR is set to 1, the MSB (bit 7) is sent, and then the TXD pin holds the states.
   If the TEIE bit in SCSCR is set to 1 at this time, a TEI interrupt request is generated.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 14.12 shows an example of SCI transmit operation.

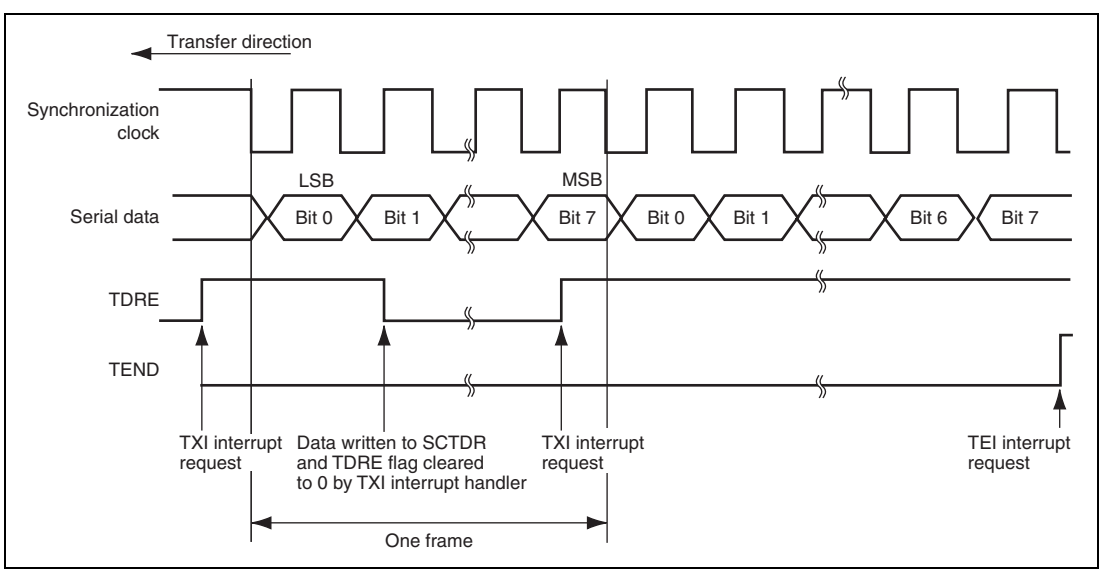
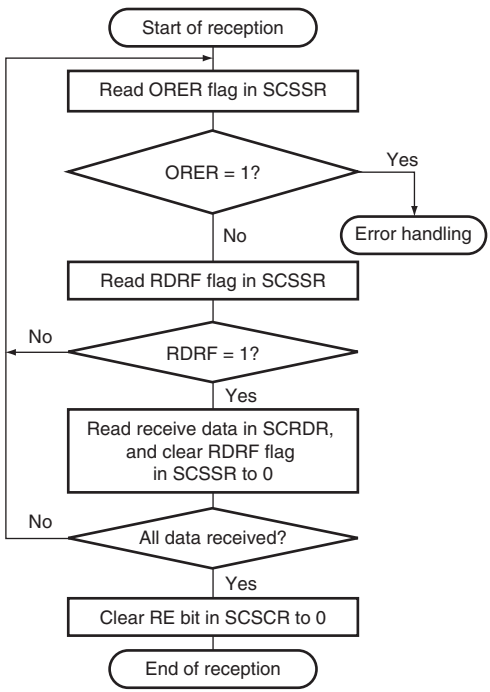



**Figure 14.12 Example of SCI Transmit Operation**

**Receiving Serial Data (Clock Synchronous Mode):** Figures 14.13 and 14.14 show a sample flowchart for receiving serial data. Use the following procedure for serial data reception after enabling the SCIF for reception.

When switching from asynchronous mode to clock synchronous mode, make sure that the ORER, PER, and FER flags are all cleared to 0. If the FER or PER flag is set to 1, the RDRF flag will not be set and data reception cannot be started.




[1] Receive error handling:

Read the ORER flag in SCSSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[2] SCI status check and receive data read:

Read SCSSR and check that RDRF = 1, then read the receive data in SCRDR, and clear the RDRF flag to 0. The transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[3] Serial reception continuation procedure:

To continue serial reception, read the receive data register (SCRDR) and clear the RDRF flag to 0 before the MSB (bit 7) of the current frame is received. The RDRF flag is cleared automatically when the data transfer controller (DTC) is activated by a receive-data-full interrupt (RXI) request to read the SCRDR value, and this step is not needed.

**Figure 14.13 Sample Flowchart for Receiving Serial Data (1)**

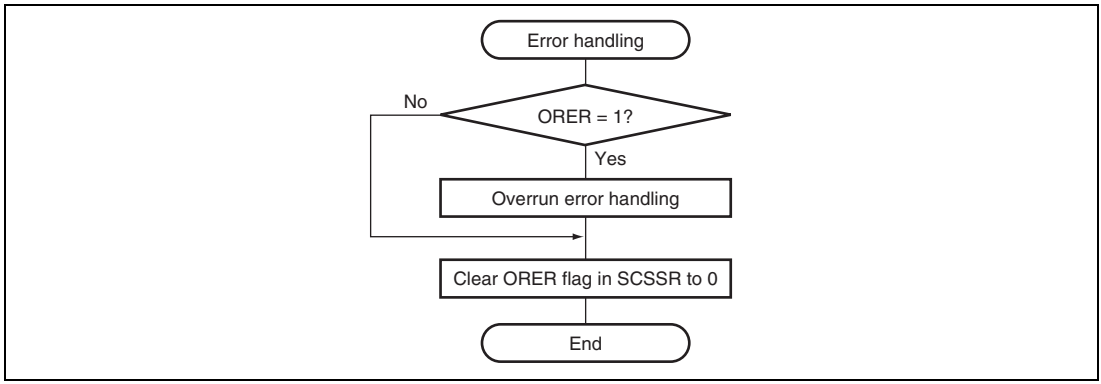



**Figure 14.14 Sample Flowchart for Receiving Serial Data (2)**

**In receiving, the SCI operates as follows:**

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After receiving the data, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from SCRSR to SCRDR. If this check is passed, the SCI sets the RDRF flag to 1 and stores the received data in SCRDR. If a receive error is detected, the SCI operates as shown in table 14.16. In this state, subsequent reception cannot be continued. In addition, the RDRF flag will not be set to 1 after reception; be sure to clear the RDRF flag to 0.
3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCSCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the RIE bit in SCSCR is also set to 1, the SCI requests a receive error interrupt (ERI).

Figure 14.15 shows an example of SCI receive operation.

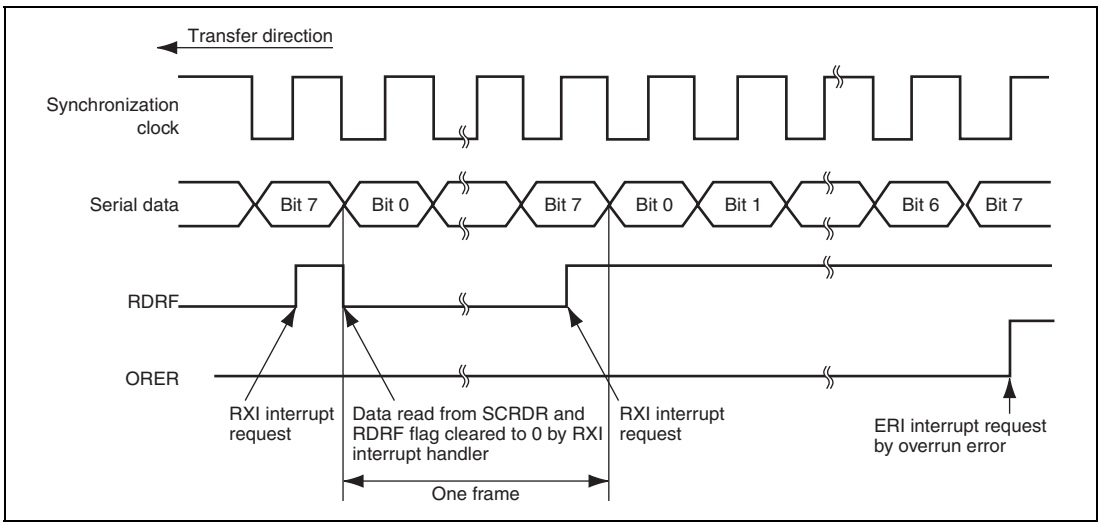
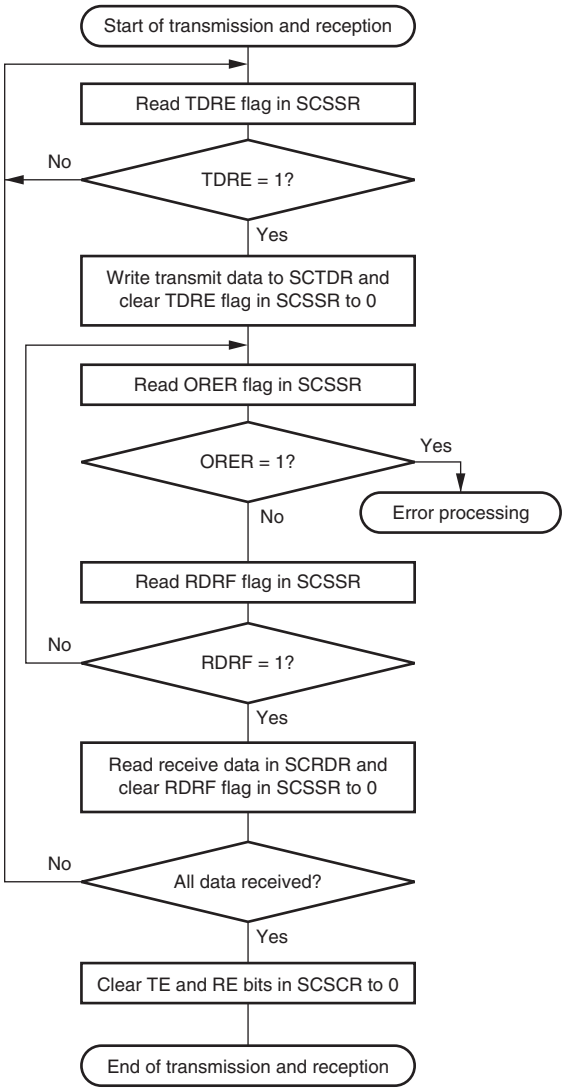



**Figure 14.15 Example of SCI Receive Operation**

**Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode):** Figure 14.16 shows a sample flowchart for transmitting and receiving serial data simultaneously.

Use the following procedure for serial data transmission and reception after enabling the SCI for transmission and reception.




[1] SCI status check and transmit data write:
Read SCSSR and check that the TDRE flag is set to 1, then write transmit data to SCTDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[2] Receive error processing:
If a receive error occurs, read the ORER flag in SCSSR, and after performing the appropriate error processing, clear the ORER flag to 0.
Reception cannot be resumed if the ORER flag is set to 1.

[3] SCI status check and receive data read:
Read SCSSR and check that the RDRF flag is set to 1, then read the receive data in SCRDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading SCRDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to SCTDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to SCTDR. Also, the RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the SCRDR value is read.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

**Figure 14.16 Sample Flowchart for Transmitting/Receiving Serial Data**

### 14.4.4 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 14.17 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. The receiving station skips data until data with a 1 multiprocessor bit is sent. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCSCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from SCRSR to SCRDR, error flag detection, and setting the SCSSR status flags, RDRF, FER, and OER to 1 are inhibited until data with a 1 multiprocessor bit is received. On reception of receive character with a 1 multiprocessor bit, the MPBR bit in SCSSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCSCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

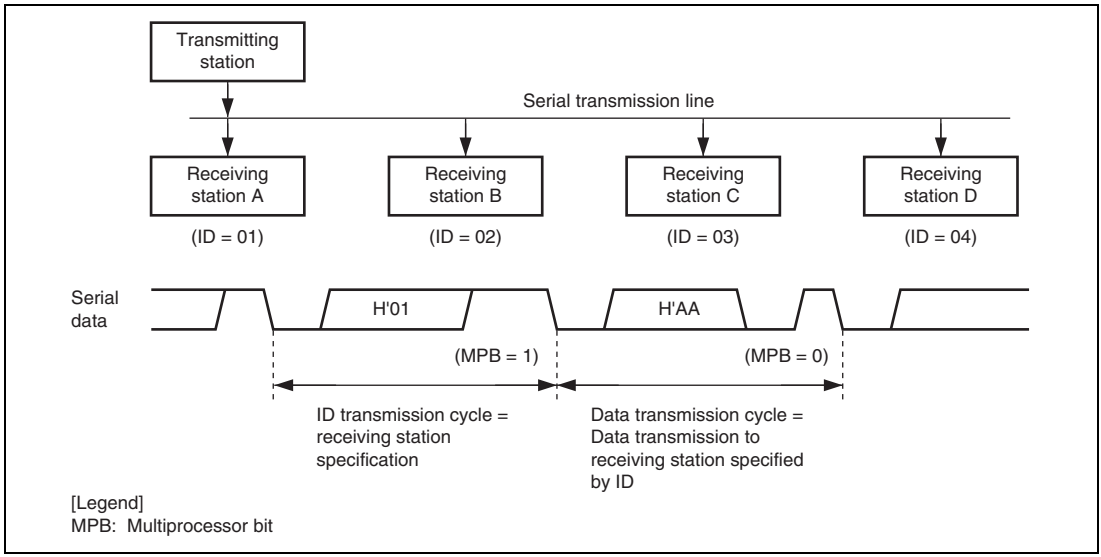
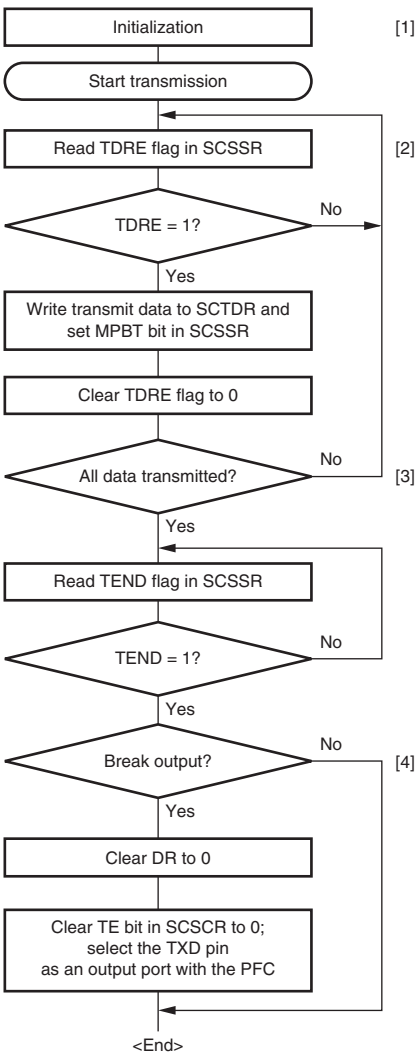



**Figure 14.17 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)**

### 14.4.5 Multiprocessor Serial Data Transmission

Figure 14.18 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SCSSR to 1 before transmission. Maintain the MPBT value at 1 until the ID transmission actually completes. For a data transmission cycle, clear the MPBT bit in SCSSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.




[1] SCI initialization:
Set the TXD pin using the PFC.
After the TE bit is set to 1, 1 is output for one frame, and transmission is enabled. However, data is not transmitted.

[2] SCI status check and transmit data write:
Read SCSSR and check that the TDRE flag is set to 1, then write transmit data to SCTDR. Set the MPBT bit in SCSSR to 0 or 1. Finally, clear the TDRE flag to 0.
To transmit an ID after the SCI is initialized, write the ID to SCTDR. The data is immediately transferred to SCTSR and the TDRE flag is set to 1. At this point the ID has not yet been transmitted from the TXD pin, so it is necessary to maintain the MPBT value at 1. Clear the MPBT bit to 0 after the next data to be transmitted is written to SCTDR and the TDRE flag is set to 1.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to SCTDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to SCTDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, first clear the port data register (DR) to 0, then clear the TE bit to 0 in SCSCR and use the PFC to select the TXD pin as an output port.

**Figure 14.18 Sample Multiprocessor Serial Transmission Flowchart**

### 14.4.6 Multiprocessor Serial Data Reception

Figures 14.20 and 14.21 show a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCSCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to SCRDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 14.19 shows an example of SCI operation for multiprocessor format reception.

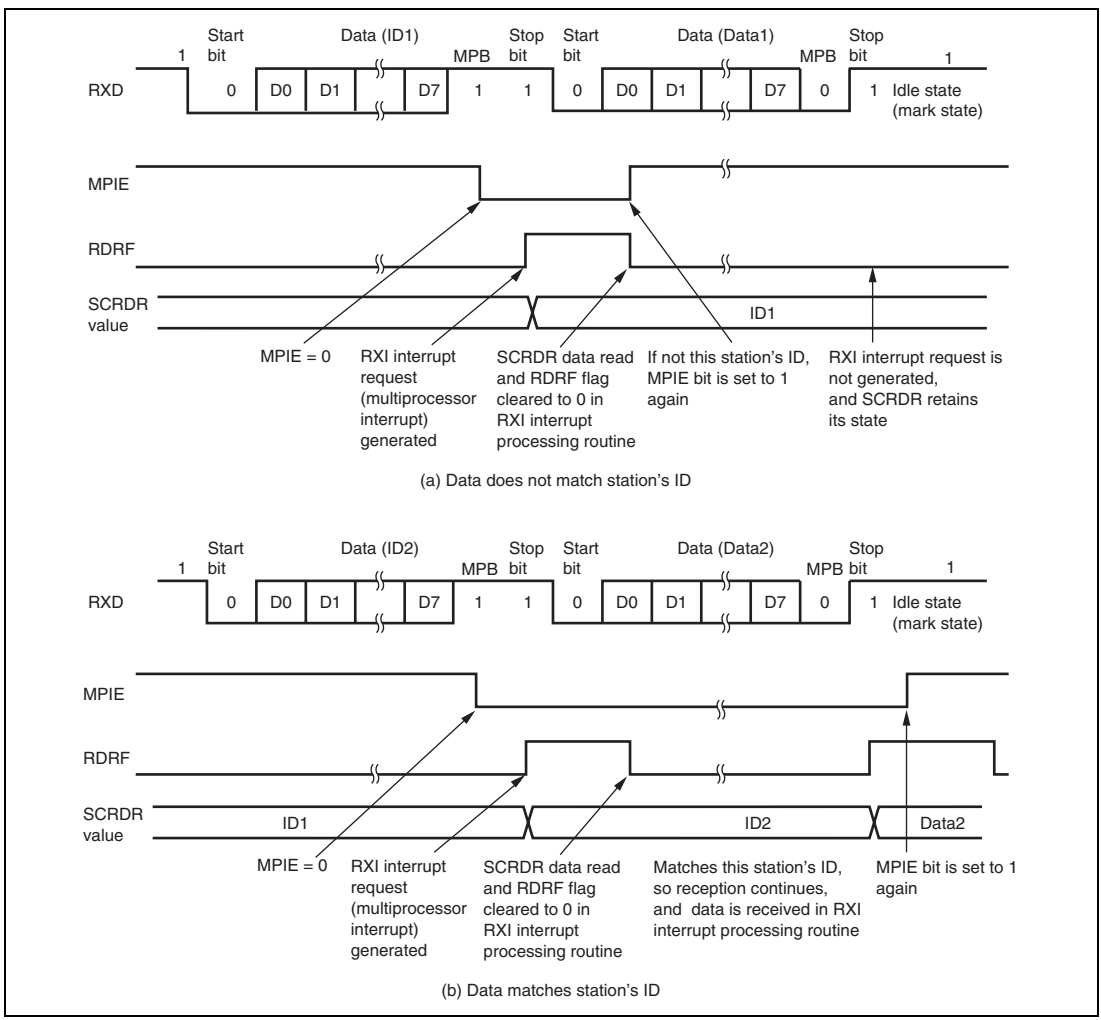
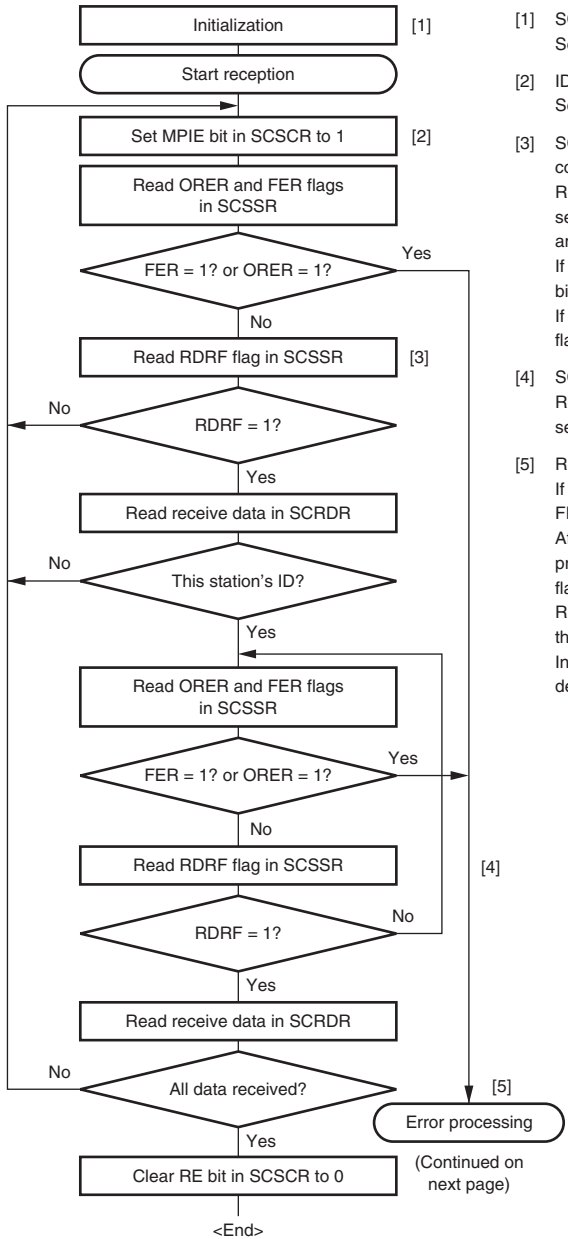



**Figure 14.19 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)**




[1] SCI initialization:
Set the RXD pin using the PFC.

[2] ID reception cycle:
Set the MPIE bit in SCSCR to 1.

[3] SCI status check, ID reception and comparison:
Read SCSSR and check that the RDRF flag is set to 1, then read the receive data in SCRDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SCSSR and check that the RDRF flag is set to 1, then read the data in SCRDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SCSSR to identify the error.
After performing the appropriate error processing, ensure that the ORER and FER flags are all cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RXD pin value.

**Figure 14.20 Sample Multiprocessor Serial Reception Flowchart (1)**

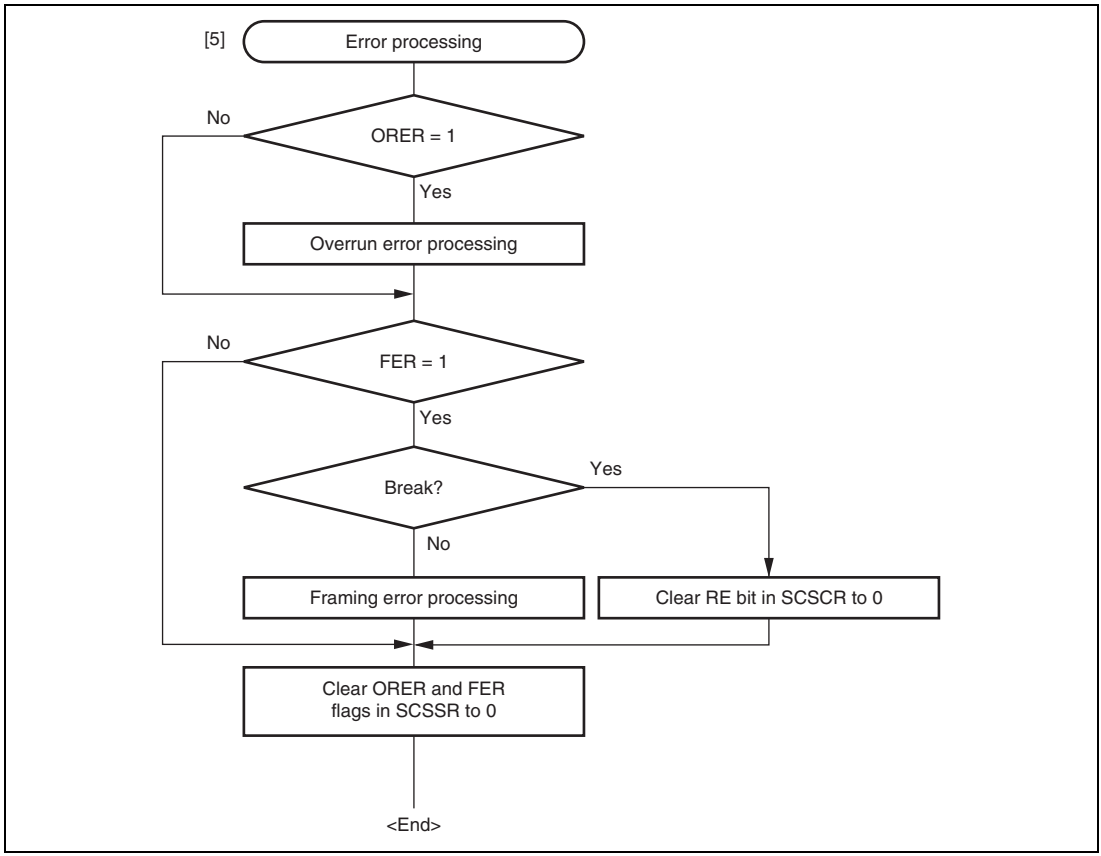



**Figure 14.21 Sample Multiprocessor Serial Reception Flowchart (2)**

## 14.5 SCI Interrupt Sources and DTC

The SCI has four interrupt sources: transmit end (TEI), receive error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI) interrupt requests.

Table 14.17 shows the interrupt sources. The interrupt sources are enabled or disabled by means of the TIE, RIE, and TEIE bits in SCSCR and the EIO bit in SCSPTR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When the TDRE flag in the serial status register (SCSSR) is set to 1, a TDR empty interrupt request is generated. This request can be used to activate the data transfer controller (DTC) to transfer data. The TDRE flag is automatically cleared to 0 when data is written to the transmit data register (SCTDR) through the DTC.

When the RDRF flag in SCSSR is set to 1, an RDR full interrupt request is generated. This request can be used to activate the DTC to transfer data. The RDRF flag is automatically cleared to 0 when data is read from the receive data register (SCRDR) through the DTC.

When the ORER, FER, or PER flag in SCSSR is set to 1, an ERI interrupt request is generated. This request cannot be used to activate the DTC. It is possible to disable generation of RXI interrupt requests and allow only ERI interrupt requests to be generated during data reception processing. To accomplish this, set the RIE bit to 1 and the EIO bit in SCSPTR to 1. Note that setting the EIO bit to 1 will prevent the DTC from transferring received data because no ERI interrupt requests are generated.

When the TEND flag in SCSSR is set to 1, a TEI interrupt request is generated. This request cannot be used to activate the DTC.

The TXI interrupt indicates that transmit data can be written, and the TEI interrupt indicates that transmission has been completed.

**Table 14.17 SCI Interrupt Sources**

| Interrupt Source | Description | DTC Activation |
|---|---|---|
| ERI | Interrupt caused by receive error (ORER, FER, or PER) | Not possible |
| RXI | Interrupt caused by receive data full (RDRF) | Possible |
| TXI | Interrupt caused by transmit data empty (TDRE) | Possible |
| TEI | Interrupt caused by transmit end (TENT) | Not possible |

## 14.6 Serial Port Register (SCSPTR) and SCI Pins

The relationship between SCSPTR and the SCI pins is shown in figures 14.22 and 14.23.

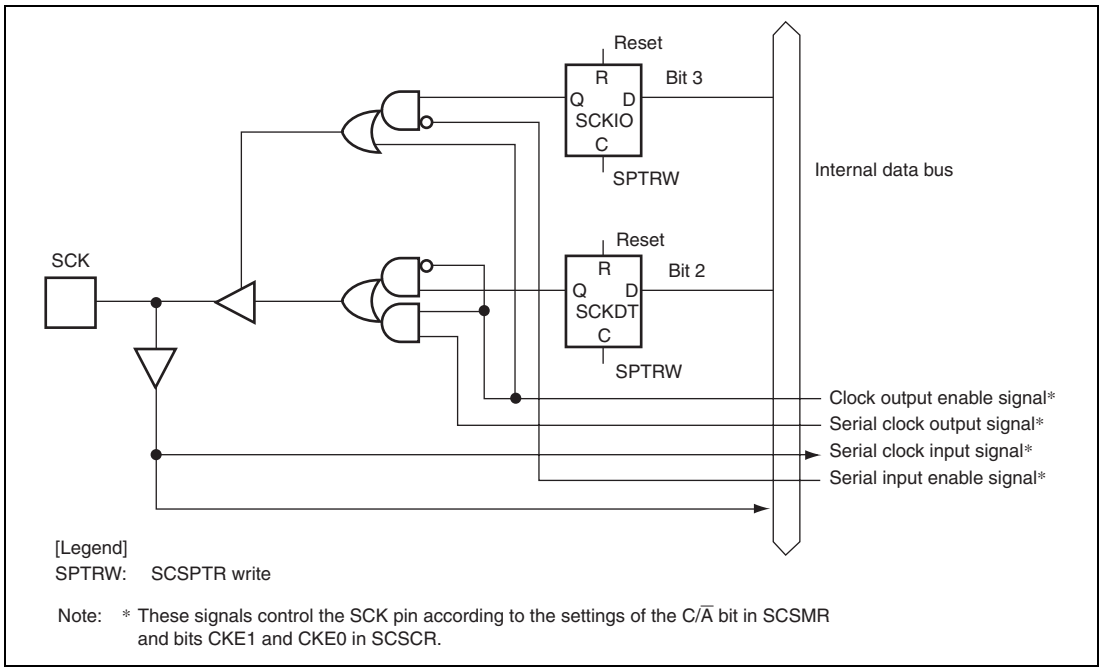



**Figure 14.22 SCKIO Bit, SCKDT Bit, and SCK Pin**

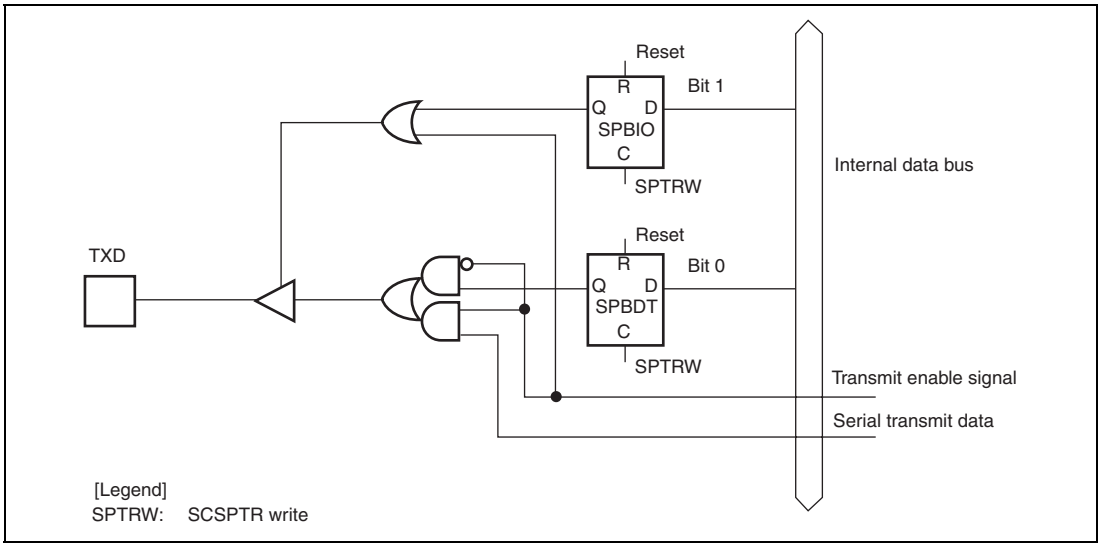



**Figure 14.23 SPBIO Bit, SPBDT Bit, and TXD Pin**

## 14.7 Usage Notes

### 14.7.1 SCTDR Writing and TDRE Flag

The TDRE flag in the serial status register (SCSSR) is a status flag indicating transferring of transmit data from SCTDR into SCTSR. The SCI sets the TDRE flag to 1 when it transfers data from SCTDR to SCTSR.

Data can be written to SCTDR regardless of the TDRE bit status.

If new data is written in SCTDR when TDRE is 0, however, the old data stored in SCTDR will be lost because the data has not yet been transferred to SCTSR. Before writing transmit data to SCTDR, be sure to check that the TDRE flag is set to 1.

### 14.7.2 Multiple Receive Error Occurrence

If multiple receive errors occur at the same time, the status flags in SCSSR are set as shown in table 14.18. When an overrun error occurs, data is not transferred from the receive shift register (SCRSR) to the receive data register (SCRDR) and the received data will be lost.

**Table 14.18 SCSSR Status Flag Values and Transfer of Received Data**

| Receive Errors Generated | SCSSR Status Flags | | | | Receive Data Transfer from SCRSR to SCRDR |
|---|---|---|---|---|---|
| | RDRF | ORER | FER | PER | |
| Overrun error | 1 | 1 | 0 | 0 | Not transferred |
| Framing error | 0 | 0 | 1 | 0 | Transferred |
| Parity error | 0 | 0 | 0 | 1 | Transferred |
| Overrun error + framing error | 1 | 1 | 1 | 0 | Not transferred |
| Overrun error + parity error | 1 | 1 | 0 | 1 | Not transferred |
| Framing error + parity error | 0 | 0 | 1 | 1 | Transferred |
| Overrun error + framing error + parity error | 1 | 1 | 1 | 1 | Not transferred |

### 14.7.3 Break Detection and Processing

Break signals can be detected by reading the RXD pin directly when a framing error (FER) is detected. In the break state the input from the RXD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCRDR is halted in the break state, the SCI receiver continues to operate.

### 14.7.4 Sending a Break Signal

The I/O condition and level of the TXD pin are determined by the SPB0IO and SPB0DT 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, TXD pin does not work. During the period, mark status is performed by SPB0DT bit. Therefore, the SPB0IO and SPB0DT bits should be set to 1 (high level output).

To send a break signal during serial transmission, clear the SPB0DT bit to 0 (low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TXD pin.

### 14.7.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCI operates on a base clock with a frequency of 16 times the transfer rate in asynchronous mode. In reception, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 14.24.

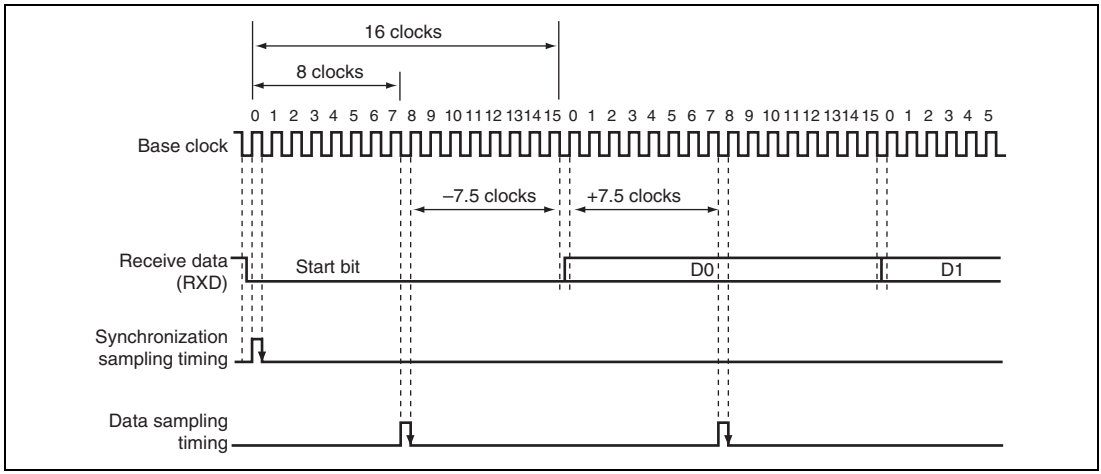



**Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode**

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

**Equation 1:**

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1+F) \right| \times 100\ \%$$

Where: M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

**Equation 2:**

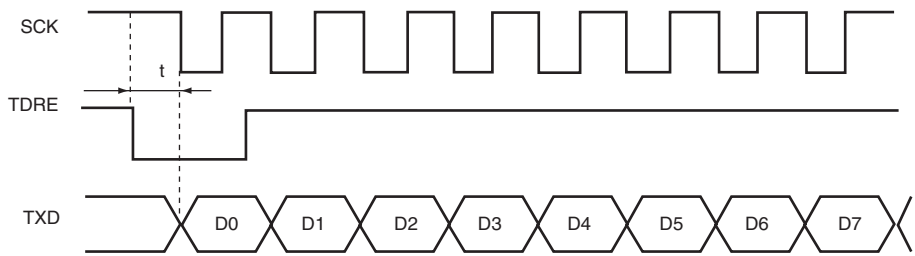
When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

### 14.7.6 Note on Using DTC

When the external clock source is used for the clock for synchronization, input the external clock after waiting for five or more cycles of the peripheral operating clock after SCTDR is modified through the DTC. If a transmit clock is input within four cycles after SCTDR is modified, a malfunction may occur (figure 14.25).




Note: When using the external clock, t must be set to larger than 4 cycles.

**Figure 14.25 Example of Clock Synchronous Transfer Using DTC**

When data is written to SCTDR by activating the DTC by a TXI interrupt, the TEND flag value becomes undefined. In this case, do not use the TEND flag as the transmit end flag.

### 14.7.7 Note on Using External Clock in Clock Synchronous Mode

TE and RE must be set to 1 after waiting for four or more cycles of the peripheral operating clock after the SCK external clock is changed from 0 to 1.

TE and RE must be set to 1 only while the SCK external clock is 1.

### 14.7.8 Module Standby Mode Setting

SCI operation can be disabled or enabled using the standby control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes.

# Section 15 Synchronous Serial Communication Unit (SSU)

This LSI has an independent synchronous serial communication unit (SSU) channel. The SSU has master mode in which this LSI outputs clocks as a master device for synchronous serial communication and slave mode in which clocks are input from an external device for synchronous serial communication. Synchronous serial communication can be performed with devices having different clock polarity and clock phase.

## 15.1 Features

- Choice of SSU mode and clock synchronous mode
- Choice of master mode and slave mode
- Choice of standard mode and bidirectional mode
- Synchronous serial communication with devices with different clock polarity and clock phase
- Choice of 8/16/32-bit width of transmit/receive data
- Full-duplex communication capability

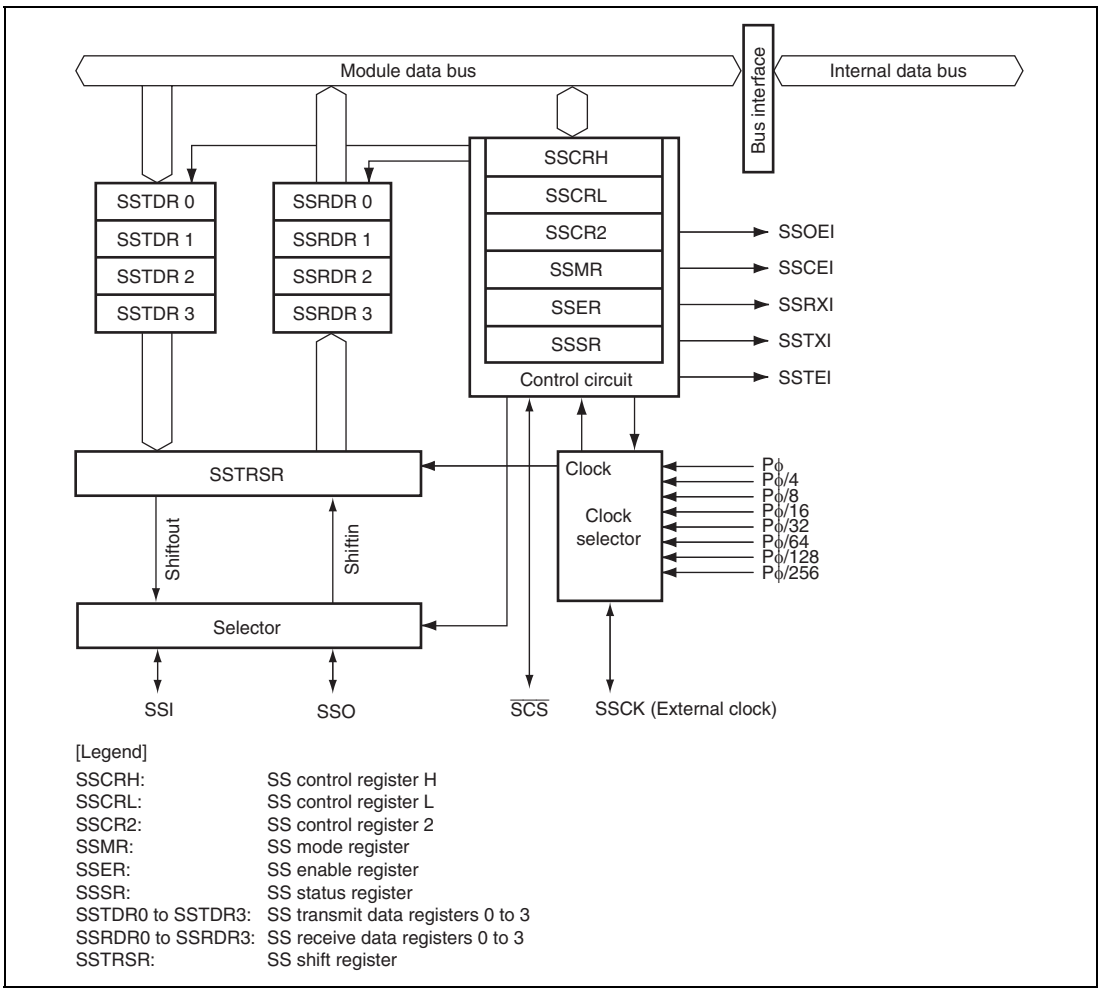
  The shift register is incorporated, enabling transmission and reception to be executed simultaneously.
- Consecutive serial communication
- Choice of LSB-first or MSB-first transfer
- Choice of a clock source

  Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/32, Pϕ/64, Pϕ/128, Pϕ/256, or an external clock
- Five interrupt sources

  Transmit end, transmit data register empty, receive data full, overrun error, and conflict error.

  The data transfer controller (DTC) can be activated by a transmit data register empty request or a receive data full request to transfer data.
- Module standby mode can be set

Figure 15.1 shows a block diagram of the SSU.




[Legend]

| | |
|---|---|
| SSCRH: | SS control register H |
| SSCRL: | SS control register L |
| SSCR2: | SS control register 2 |
| SSMR: | SS mode register |
| SSER: | SS enable register |
| SSSR: | SS status register |
| SSTDR0 to SSTDR3: | SS transmit data registers 0 to 3 |
| SSRDR0 to SSRDR3: | SS receive data registers 0 to 3 |
| SSTRSR: | SS shift register |

**Figure 15.1 Block Diagram of SSU**

## 15.2 Input/Output Pins

Table 15.1 shows the SSU pin configuration.

**Table 15.1 Pin Configuration**

| Symbol | I/O | Function |
|---|---|---|
| SSCK | I/O | SSU clock input/output |
| SSI | I/O | SSU data input/output |
| SSO | I/O | SSU data input/output |
| $\overline{\text{SCS}}$ | I/O | SSU chip select input/output |

## 15.3 Register Descriptions

The SSU has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 25, List of Registers.

**Table 15.2 Register Configuration**

| Register Name | Abbreviation | R/W | Initial value | Address | Access Size |
|---|---|---|---|---|---|
| SS control register H | SSCRH | R/W | H'0D | H'FFFFCD00 | 8, 16 |
| SS control register L | SSCRL | R/W | H'00 | H'FFFFCD01 | 8 |
| SS mode register | SSMR | R/W | H'00 | H'FFFFCD02 | 8, 16 |
| SS enable register | SSER | R/W | H'00 | H'FFFFCD03 | 8 |
| SS status register | SSSR | R/W | H'04 | H'FFFFCD04 | 8, 16 |
| SS control register 2 | SSCR2 | R/W | H'00 | H'FFFFCD05 | 8 |
| SS transmit data register 0 | SSTDR0 | R/W | H'00 | H'FFFFCD06 | 8, 16 |
| SS transmit data register 1 | SSTDR1 | R/W | H'00 | H'FFFFCD07 | 8 |
| SS transmit data register 2 | SSTDR2 | R/W | H'00 | H'FFFFCD08 | 8, 16 |
| SS transmit data register 3 | SSTDR3 | R/W | H'00 | H'FFFFCD09 | 8 |
| SS receive data register 0 | SSRDR0 | R | H'00 | H'FFFFCD0A | 8, 16 |
| SS receive data register 1 | SSRDR1 | R | H'00 | H'FFFFCD0B | 8 |
| SS receive data register 2 | SSRDR2 | R | H'00 | H'FFFFCD0C | 8, 16 |
| SS receive data register 3 | SSRDR3 | R | H'00 | H'FFFFCD0D | 8 |

### 15.3.1 SS Control Register H (SSCRH)

SSCRH specifies master/slave device selection, bidirectional mode enable, SSO pin output value selection, SSCK pin selection, and $\overline{SCS}$ pin selection.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MSS | BIDE | - | SOL | SOLP | - | CSS[1:0] | |
| Initial value: | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| R/W: | R/W | R/W | R | R/W | R/W | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | MSS | 0 | R/W | Master/Slave Device Select<br>Selects that this module is used in master mode or slave mode. When master mode is selected, transfer clocks are output from the SSCK pin. When the CE bit in SSSR is set, this bit is automatically cleared.<br>0: Slave mode is selected.<br>1: Master mode is selected. |
| 6 | BIDE | 0 | R/W | Bidirectional Mode Enable<br>Selects that both serial data input pin and output pin are used or one of them is used. However, transmission and reception are not performed simultaneously when bidirectional mode is selected. For details, section 15.4.3, Relationship between Data Input/Output Pins and Shift Register.<br>0: Standard mode (two pins are used for data input and output)<br>1: Bidirectional mode (one pin is used for data input and output) |
| 5 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | SOL | 0 | R/W | Serial Data Output Value Select<br>The serial data output retains its level of the last bit after completion of transmission. The output level before or after transmission can be specified by setting this bit. When specifying the output level, use the MOV instruction after clearing the SOLP bit to 0. Since writing to this bit during data transmission causes malfunctions, this bit should not be changed.<br>0: Serial data output is changed to low.<br>1: Serial data output is changed to high. |
| 3 | SOLP | 1 | R/W | SOL Bit Write Protect<br>When changing the output level of serial data, set the SOL bit to 1 or clear the SOL bit to 0 after clearing the SOLP bit to 0 using the MOV instruction.<br>0: Output level can be changed by the SOL bit<br>1: Output level cannot be changed by the SOL bit. This bit is always read as 1. |
| 2 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 1, 0 | CSS[1:0] | 01 | R/W | $\overline{\text{SCS}}$ Pin Select<br>Select that the $\overline{\text{SCS}}$ pin functions as $\overline{\text{SCS}}$ input or output.<br>00: Setting prohibited<br>01: Setting prohibited<br>10: Function as $\overline{\text{SCS}}$ automatic input/output (function as $\overline{\text{SCS}}$ input before and after transfer and output a low level during transfer)<br>11: Function as $\overline{\text{SCS}}$ automatic output (outputs a high level before and after transfer and outputs a low level during transfer) |

### 15.3.2 SS Control Register L (SSCRL)

SSCRL selects operating mode, software reset, and transmit/receive data length.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FCLRM | SSUMS | SRES | - | - | - | DATS[1:0] | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | FCLRM | 0 | R/W | Flag Clear Mode<br>Selects whether the SSRXI and SSTXI interrupt flags are cleared on writing to SSTDR or reading from SSRDR or on completion of DTC transfer. When using the DTC, set this bit to 0.<br>0: Flags are cleared when DTC transfer is completed (except when transfer counter value is H'0000)<br>1: Flags are cleared on SSTDR or SSRDR access |
| 6 | SSUMS | 0 | R/W | Selects transfer mode from SSU mode and clock synchronous mode.<br>0: SSU mode<br>1: Clock synchronous mode |
| 5 | SRES | 0 | R/W | Software Reset<br>Setting this bit to 1 forcibly resets the SSU internal sequencer. After that, this bit is automatically cleared. The ORER, TEND, TDRE, RDRF, and CE bits in SSSR and the TE and RE bits in SSER are also initialized. Values of other bits for SSU registers are held.<br>To stop transfer, set this bit to 1 to reset the SSU internal sequencer. |
| 4 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1, 0 | DATS[1:0] | 00 | R/W | Transmit/Receive Data Length Select<br>Select serial data length.<br>00: 8 bits<br>01: 16 bits<br>10: 32 bits<br>11: Setting prohibited |

### 15.3.3 SS Mode Register (SSMR)

SSMR selects the MSB first/LSB first, clock polarity, clock phase, and clock rate of synchronous serial communication.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MLS | CPOS | CPHS | - | - | CKS[2:0] | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 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 | MLS | 0 | R/W | MSB First/LSB First Select<br>Selects that the serial data is transmitted in MSB first or LSB first.<br>0: LSB first<br>1: MSB first |
| 6 | CPOS | 0 | R/W | Clock Polarity Select<br>Selects the SSCK clock polarity.<br>0: High output in idle mode, and low output in active mode<br>1: Low output in idle mode, and high output in active mode |
| 5 | CPHS | 0 | R/W | Clock Phase Select (Only for SSU Mode)<br>Selects the SSCK clock phase.<br>0: Data changes at the first edge.<br>1: Data is latched at the first edge. |
| 4, 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 to 0 | CKS[2:0] | 000 | R/W | Transfer Clock Rate Select<br>Select the transfer clock rate (prescaler division rate) when an internal clock is selected.<br>000: Reserved<br>001: Pϕ/4<br>010: Pϕ/8<br>011: Pϕ/16<br>100: Pϕ/32<br>101: Pϕ/64<br>110: Pϕ/128<br>111: Pϕ/256 |

### 15.3.4 SS Enable Register (SSER)

SSER performs transfer/receive control of synchronous serial communication and setting of interrupt enable.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TE | RE | - | - | TEIE | TIE | RIE | CEIE |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 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 | TE | 0 | R/W | Transmit Enable<br>When this bit is set to 1, transmission is enabled. |
| 6 | RE | 0 | R/W | Receive Enable<br>When this bit is set to 1, reception is enabled. |
| 5, 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 3 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br>When this bit is set to 1, a SSTEI interrupt request is enabled. |
| 2 | TIE | 0 | R/W | Transmit Interrupt Enable<br>When this bit is set to 1, a SSTXI interrupt request is enabled. |
| 1 | RIE | 0 | R/W | Receive Interrupt Enable<br>When this bit is set to 1, an SSRXI interrupt request and an SSOEI interrupt request are enabled. |
| 0 | CEIE | 0 | R/W | Conflict Error Interrupt Enable<br>When this bit is set to 1, a SSCEI interrupt request is enabled. |

### 15.3.5 SS Status Register (SSSR)

SSSR is a status flag register for interrupts.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | ORER | - | - | TEND | TDRE | RDRF | CE |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W: | R | R/W | R | R | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | ORER | 0 | R/W | Overrun Error<br>If the next data is received while RDRF = 1, an overrun error occurs, indicating abnormal termination. SSRDR stores 1-frame receive data before an overrun error occurs and loses data to be received later. While ORER = 1, consecutive serial reception cannot be continued. Serial transmission cannot be continued, either.<br>[Setting condition]<br>• When one byte of the next reception is completed with RDRF = 1<br>[Clearing condition]<br>• When writing 0 after reading ORER = 1 |
| 5, 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | TEND | 0 | R/W | Transmit End<br>[Setting conditions]<br>• When the last bit of transmit data is transmitted while the TENDSTS bit in SSCR2 is cleared to 0 and the TDRE bit is set to 1<br>• After the last bit of transmit data is transmitted while the TENDSTS bit in SSCR2 is set to 1 and the TDRE bit is set to 1<br>[Clearing conditions]<br>• When writing 0 after reading TEND = 1<br>• When writing data to SSTDR |
| 2 | TDRE | 1 | R/W | Transmit Data Empty<br>Indicates whether or not SSTDR contains transmit data.<br>[Setting conditions]<br>• When the TE bit in SSER is 0<br>• When data is transferred from SSTDR to SSTRSR and SSTDR is ready to be written to.<br>[Clearing conditions]<br>• When writing 0 after reading TDRE = 1<br>• When writing data to SSTDR with TE = 1<br>• When the DTC is activated by an SSTXI interrupt and transmit data is written to SSTDR while the DISEL bit in MRB of the DTC is 0 (except when DTC transfer counter value is H'0000) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | RDRF | 0 | R/W | Receive Data Register Full<br>Indicates whether or not SSRDR contains receive data.<br>[Setting condition]<br>• When receive data is transferred from SSTRSR to SSRDR after successful serial data reception<br>[Clearing conditions]<br>• When writing 0 after reading RDRF = 1<br>• When reading receive data from SSRDR<br>• When the DTC is activated by an SSRXI interrupt and receive data is read into SSRDR while the DISEL bit in MRB of the DTC is 0 (except when DTC transfer counter value is H'0000) |
| 0 | CE | 0 | R/W | Conflict/Incomplete Error<br>Indicates that a conflict error has occurred when 0 is externally input to the $\overline{SCS}$ pin with SSUMS = 0 (SSU mode) and MSS = 1 (master mode).<br>If the $\overline{SCS}$ pin level changes to 1 with SSUMS = 0 (SSU mode) and MSS = 0 (slave mode), an incomplete error occurs because it is determined that a master device has terminated the transfer. In addition, when SSUMS = 0 (SSU mode) and MSS = 0 (slave mode) and the next serial receive operation starts while RDRF = 1, an incomplete error occurs even if the data received from SSRDR is read before the completion of reception and RDRF is cleared to 0 before the $\overline{SCS}$ pin is set to 1. Data reception does not continue while the CE bit is set to 1. Serial transmission also does not continue. Reset the SSU internal sequencer by setting the SRES bit in SSCRL to 1 before resuming transfer after incomplete error.<br>[Setting conditions]<br>• When a low level is input to the $\overline{SCS}$ pin in master mode (the MSS bit in SSCRH is set to 1)<br>• When the $\overline{SCS}$ pin is changed to 1 during transfer in slave mode (the MSS bit in SSCRH is cleared to 0)<br>• When in slave mode (MSS = 0 in SSCRH), the next serial receive operation starts while RDRF = 1, and data is read from SSRDR before the completion of reception, after which the $\overline{SCS}$ pin is set to 1<br>[Clearing condition]<br>• When writing 0 after reading CE = 1 |

### 15.3.6 SS Control Register 2 (SSCR2)

SSCR2 is a register that selects the assert timing of the $\overline{SCS}$ pin, data output timing of the SSO pin, and set timing of the TEND bit.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | TENDSTS | SCSATS | SSODTS | - | - |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R/W | R/W | R/W | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 4 | TENDSTS | 0 | R/W | Selects the timing of setting the TEND bit (valid in SSU and master mode).<br>0: Sets the TEND bit when the last bit is being transmitted<br>1: Sets the TEND bit after the last bit is transmitted |
| 3 | SCSATS | 0 | R/W | Selects the assertion timing of the $\overline{SCS}$ pin (valid in SSU and master mode).<br>0: Min. values of $t_{LEAD}$ and $t_{LAG}$ are $1/2 \times t_{SUcyc}$<br>1: Min. values of $t_{LEAD}$ and $t_{LAG}$ are $3/2 \times t_{SUcyc}$ |
| 2 | SSODTS | 0 | R/W | Selects the data output timing of the SSO pin (valid in SSU and master mode)<br>0: While BIDE = 0, MSS = 1, and TE = 1 or while BIDE = 1, TE = 1, and RE = 0, the SSO pin outputs data<br>1: While BIDE = 0, MSS = 1, and TE = 1 or while BIDE = 1, TE = 1, and RE = 0, the SSO pin outputs data while the $\overline{SCS}$ pin is driven low |
| 1, 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

### 15.3.7 SS Transmit Data Registers 0 to 3 (SSTDR0 to SSTDR3)

SSTDR is an 8-bit register that stores transmit data. When 8-bit data length is selected by bits DATS1 and DATS0 in SSCRL, SSTDR0 is valid. When 16-bit data length is selected, SSTDR0 and SSTDR1 are valid. When 32-bit data length is selected, SSTDR0 to SSTDR3 are valid. Do not access SSTDR that is not valid.

When the SSU detects that SSTRSR is empty, it transfers the transmit data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit data has already been written to SSTDR during serial transmission, the SSU performs consecutive serial transmission.

Although SSTDR can always be read from or written to by the CPU and DTC, to achieve reliable serial transmission, write transmit data to SSTDR after confirming that the TDRE bit in SSSR is set to 1.

| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | | All 0 | R/W | Serial transmit data |

**Table 15.3 Setting of DATS Bits in SSCRL and Corresponding SSTDR**

| | DATS[1:0] Setting | | | |
|---|---|---|---|---|
| | **00** | **01** | **10** | **11 (Invalid setting)** |
| SSTDR0 | Valid | Valid | Valid | Invalid |
| SSTDR1 | Invalid | Valid | Valid | Invalid |
| SSTDR2 | Invalid | Invalid | Valid | Invalid |
| SSTDR3 | Invalid | Invalid | Valid | Invalid |

### 15.3.8 SS Receive Data Registers 0 to 3 (SSRDR0 to SSRDR3)

SSRDR is an 8-bit register that stores receive data. When 8-bit data length is selected by bits DATS1 and DATS0 in SSCRL, SSRDR0 is valid. When 16-bit data length is selected, SSRDR0 and SSRDR1 are valid. When 32-bit data length is selected, SSRDR0 to SSRDR3 are valid. Do not access SSRDR that is not valid.

When the SSU has received 1-byte data, it transfers the received serial data from SSTRSR to SSRDR where it is stored. After this, SSTRSR is ready for reception. Since SSTRSR and SSRDR function as a double buffer in this way, consecutive receive operations can be performed.

Read SSRDR after confirming that the RDRF bit in SSSR is set to 1.

SSRDR is a read-only register, therefore, cannot be written to by the CPU.

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | | All 0 | R | Serial receive data |

**Table 15.4 Setting of DATS Bit in SSCRL and Corresponding SSRDR**

| | DATS[1:0] Setting | | | |
|---|---|---|---|---|
| | **00** | **01** | **10** | **11 (Invalid setting)** |
| SSRDR0 | Valid | Valid | Valid | Invalid |
| SSRDR1 | Invalid | Valid | Valid | Invalid |
| SSRDR2 | Invalid | Invalid | Valid | Invalid |
| SSRDR3 | Invalid | Invalid | Valid | Invalid |

### 15.3.9 SS Shift Register (SSTRSR)

SSTRSR is a shift register that transmits and receives serial data.

When data is transferred from SSTDR to SSTRSR, bit 0 of transmit data is bit 0 in the SSTDR contents (MLS = 0: LSB first communication) and is bit 7 in the SSTDR contents (MLS = 1: MSB first communication). The SSU transfers data from the LSB (bit 0) in SSTRSR to the SSO pin to perform serial data transmission.

In reception, the SSU sets serial data that has been input via the SSI pin in SSTRSR from the LSB (bit 0). When 1-byte data has been received, the SSTRSR contents are automatically transferred to SSRDR. SSTRSR cannot be directly accessed by the CPU.

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

## 15.4 Operation

### 15.4.1 Transfer Clock

A transfer clock can be selected from seven internal clocks and an external clock. Before using this module, enable the SSCK pin function in the PFC. When the MSS bit in SSCRH is 1, an internal clock is selected and the SSCK pin is used as an output pin. When transfer is started, the clock with the transfer rate set by bits CKS2 to CKS0 in SSMR is output from the SSCK pin. When MSS = 0, an external clock is selected and the SSCK pin is used as an input pin.

### 15.4.2 Relationship of Clock Phase, Polarity, and Data

The relationship of clock phase, polarity, and transfer data depends on the combination of the CPOS and CPHS bits in SSMR when the value of the SSUMS bit in SSCRL is 0. Figure 15.2 shows the relationship. When SSUMS = 1, the CPHS setting is invalid although the CPOS setting is valid.

Setting the MLS bit in SSMR selects that MSB or LSB first communication. When MLS = 0, data is transferred from the LSB to the MSB. When MLS = 1, data is transferred from the MSB to the LSB.

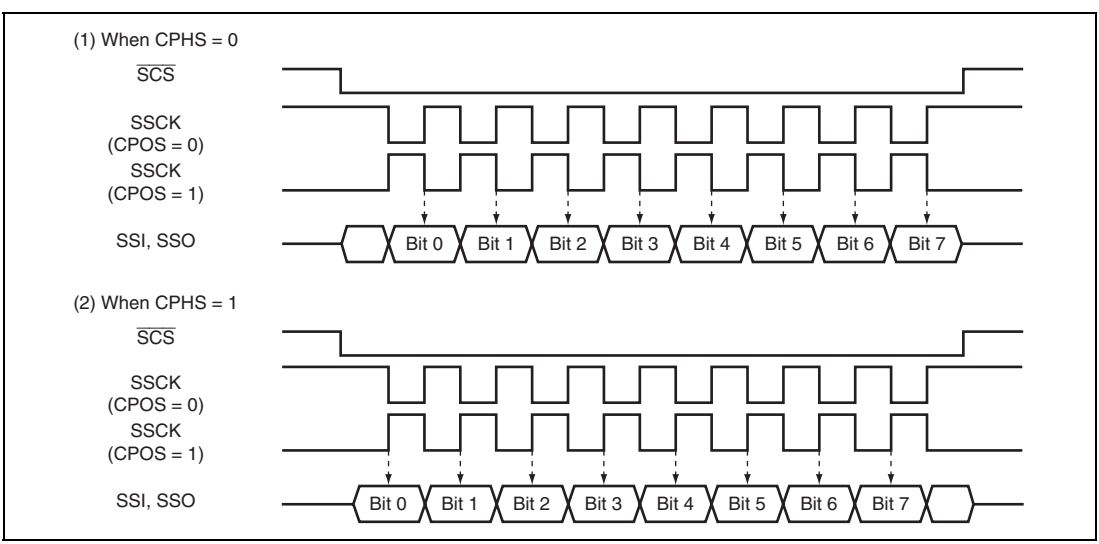



**Figure 15.2 Relationship of Clock Phase, Polarity, and Data**

### 15.4.3 Relationship between Data Input/Output Pins and Shift Register

The connection between data input/output pins and the SS shift register (SSTRSR) depends on the combination of the MSS and BIDE bits in SSCRH and the SSUMS bit in SSCRL. Figure 15.3 shows the relationship.

The SSU transmits serial data from the SSO pin and receives serial data from the SSI pin when operating with BIDE = 0 and MSS = 1 (standard, master mode) (see figure 15.3 (1)). The SSU transmits serial data from the SSI pin and receives serial data from the SSO pin when operating with BIDE = 0 and MSS = 0 (standard, slave mode) (see figure 15.3 (2)).

The SSU transmits and receives serial data from the SSO pin regardless of master or slave mode when operating with BIDE = 1 (bidirectional mode) (see figures 15.3 (3) and (4)).

However, even if both the TE and RE bits are set to 1, transmission and reception are not performed simultaneously. Either the TE or RE bit must be selected.

The SSU transmits serial data from the SSO pin and receives serial data from the SSI pin when operating with SSUMS = 1. The SSCK pin outputs the internal clock when MSS = 1 and function as an input pin when MSS = 0 (see figures 15.3 (5) and (6)).

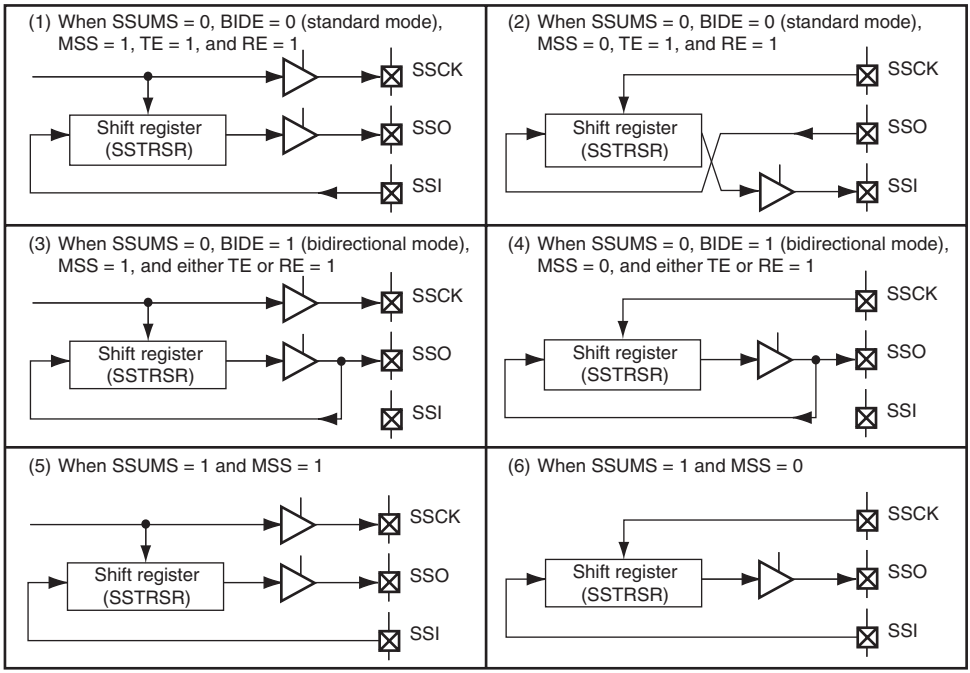



**Figure 15.3 Relationship between Data Input/Output Pins and the Shift Register**

### 15.4.4 Communication Modes and Pin Functions

The SSU switches the input/output pin (SSI, SSO, SSCK, and $\overline{SCS}$) functions according to the communication modes and register settings. The input/output directions of the pins should be selected in the port I/O registers. The relationship of communication modes and input/output pin functions are shown in tables 15.5 to 15.7.

**Table 15.5 Communication Modes and Pin States of SSI and SSO Pins**

| Communication Mode | Register Setting | | | | | Pin State | |
|---|---|---|---|---|---|---|---|
| | SSUMS | BIDE | MSS | TE | RE | SSI | SSO |
| SSU communication mode | 0 | 0 | 0 | 0 | 1 | — | Input |
| | | | | 1 | 0 | Output | — |
| | | | | | 1 | Output | Input |
| | | | 1 | 0 | 1 | Input | — |
| | | | | 1 | 0 | — | Output |
| | | | | | 1 | Input | Output |
| SSU (bidirectional) communication mode | 0 | 1 | 0 | 0 | 1 | — | Input |
| | | | | 1 | 0 | — | Output |
| | | | 1 | 0 | 1 | — | Input |
| | | | | 1 | 0 | — | Output |
| Clock synchronous communication mode | 1 | 0 | 0 | 0 | 1 | Input | — |
| | | | | 1 | 0 | — | Output |
| | | | | | 1 | Input | Output |
| | | | 1 | 0 | 1 | Input | — |
| | | | | 1 | 0 | — | Output |
| | | | | | 1 | Input | Output |

[Legend]

—: Not used as SSU pin

**Table 15.6 Communication Modes and Pin States of SSCK Pin**

| Communication Mode | Register Setting | | Pin State |
|---|---|---|---|
| | SSUMS | MSS | SSCK |
| SSU communication mode | 0 | 0 | Input |
| | | 1 | Output |
| Clock synchronous communication mode | 1 | 0 | Input |
| | | 1 | Output |

[Legend]

—: Not used as SSU pin

**Table 15.7 Communication Modes and Pin States of $\overline{SCS}$ Pin**

| Communication Mode | Register Setting | | | | Pin State |
|---|---|---|---|---|---|
| | SSUMS | MSS | CSS1 | CSS0 | $\overline{SCS}$ |
| SSU communication mode | 0 | 0 | x | x | Input |
| | | 1 | 0 | 0 | — |
| | | | 0 | 1 | — |
| | | | 1 | 0 | Automatic input/output |
| | | | 1 | 1 | Output |
| Clock synchronous communication mode | 1 | x | x | x | — |

[Legend]

x: Don't care

—: Not used as SSU pin

### 15.4.5 SSU Mode

In SSU mode, data communications are performed via four lines: clock line (SSCK), data input line (SSI or SSO), data output line (SSI or SSO), and chip select line ($\overline{SCS}$).

In addition, the SSU supports bidirectional mode in which a single pin functions as data input and data output lines.

#### (1) Initial Settings in SSU Mode

Figure 15.4 shows an example of the initial settings in SSU mode. Before data transfer, clear both the TE and RE bits in SSER to 0 to set the initial values.

Note: Before changing operating modes and communications formats, clear both the TE and RE bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the RE bit to 0 does not change the values of the RDRF and ORER bits and SSRDR. Those bits retain the previous values.

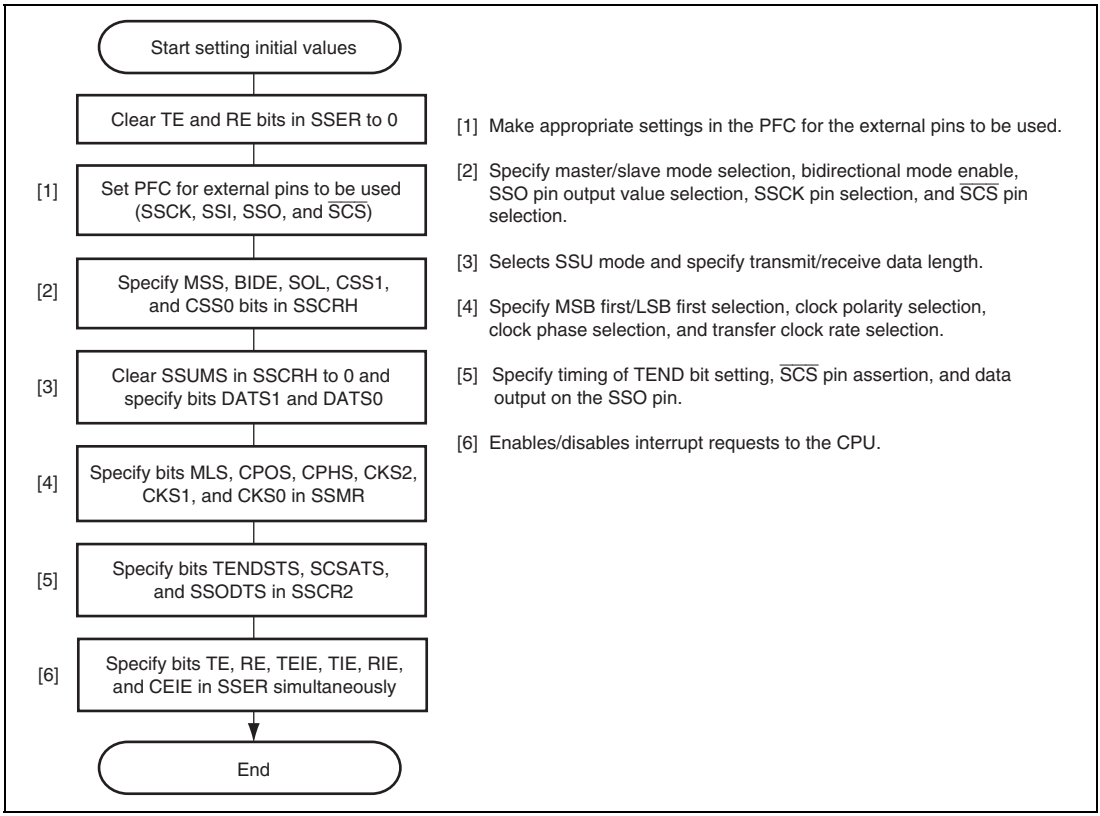



**Figure 15.4 Example of Initial Settings in SSU Mode**

### (2) Data Transmission

Figure 15.5 shows an example of transmission operation, and figure 15.6 shows a flowchart example of data transmission.

When transmitting data, the SSU operates as shown below.

In master mode, the SSU outputs a transfer clock and data. In slave mode, when a low level signal is input to the $\overline{\text{SCS}}$ pin and a transfer clock is input to the SSCK pin, the SSU outputs data in synchronization with the transfer clock.

Writing transmit data to SSTDR after the TE bit is set to 1 clears the TDRE bit in SSSR to 0, and the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit to 1 and starts transmission. At this time, if the TIE bit in SSER is set to 1, a TXI interrupt is generated.

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transferred to SSTRSR to start the next frame transmission. When the 8th bit of transmit data has been transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. At this time, if the TEIE bit is set to 1, a TEI interrupt is generated. After transmission, the output level of the SSCK pin is fixed high when CPOS = 0 and low when CPOS = 1.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the ORER bit is cleared to 0 before transmission.

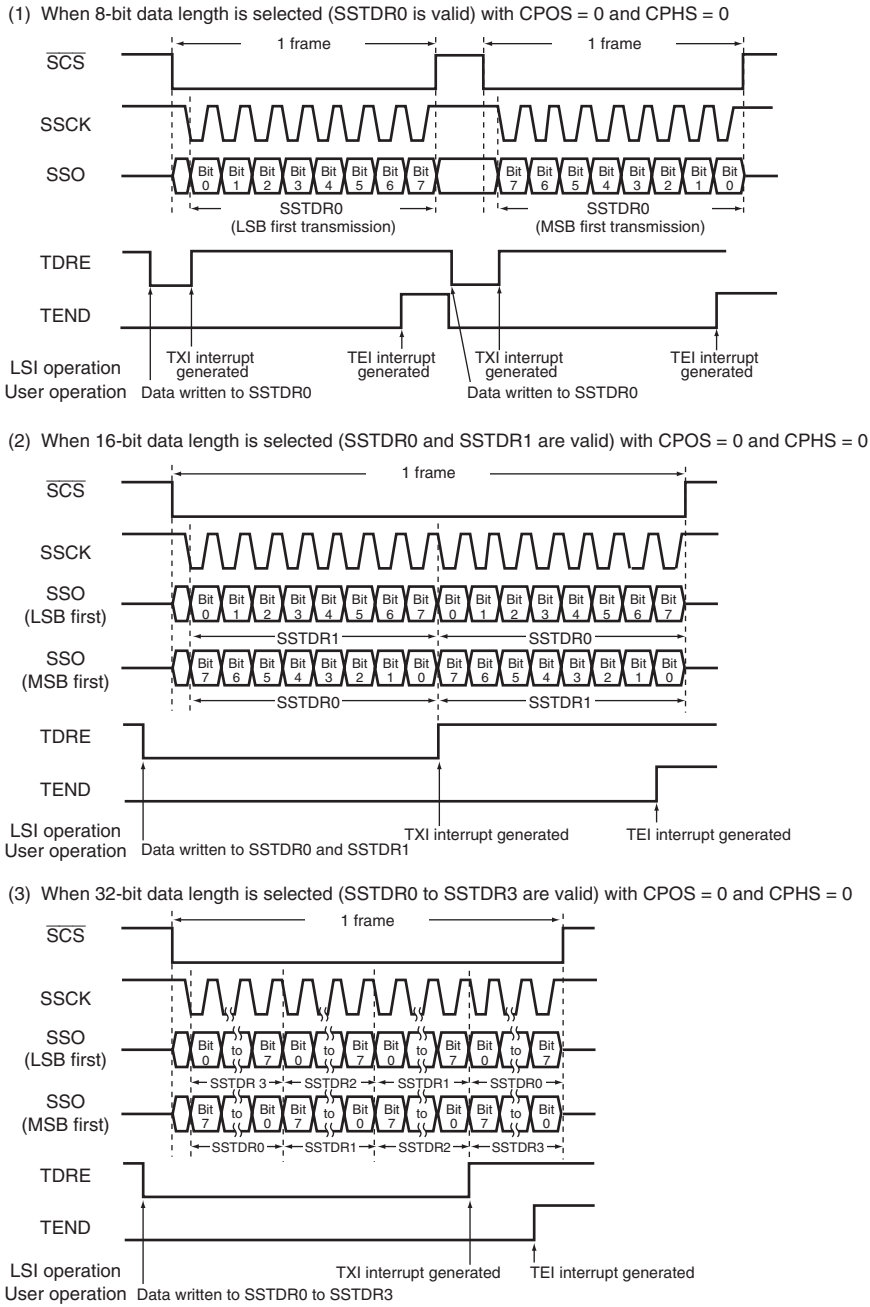
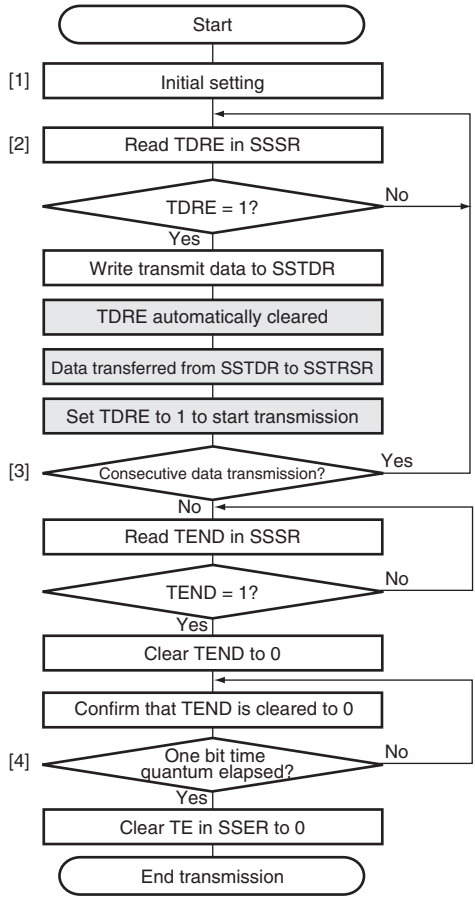



**Figure 15.5 Example of Transmission Operation (SSU Mode)**




[1] Initial setting:
Specify the transmit data format.

[2] Check that the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Procedure for consecutive data transmission:
To continue data transmission, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

[4] Procedure for data transmission end:
To end data transmission, confirm that the TEND bit is cleared to 0. After completion of transmitting the last bit, clear the TE bit to 0.

Note: Hatching boxes represent SSU internal operations.

**Figure 15.6 Flowchart Example of Data Transmission (SSU Mode)**

### (3) Data Reception

Figure 15.7 shows an example of reception operation, and figure 15.8 shows a flowchart example of data reception. When receiving data, the SSU operates as shown below.

After setting the RE bit to 1 and dummy-reading SSRDR, the SSU starts data reception.

In master mode, the SSU outputs a transfer clock and receives data. In slave mode, when a low level signal is input to the $\overline{\text{SCS}}$ pin and a transfer clock is input to the SSCK pin, the SSU receives data in synchronization with the transfer clock.

When 1-frame data has been received, the RDRF bit in SSSR is set to 1 and the receive data is stored in SSRDR. At this time, if the RIE bit in SSER is set to 1, an RXI interrupt is generated. The RDRF bit is automatically cleared to 0 by reading SSRDR.

When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORER bit in SSSR is set to 1. This indicates that an overrun error (OEI) has occurred. At this time, data reception is stopped. While the ORER bit in SSSR is set to 1, reception is not performed. To resume the reception, clear the ORER bit to 0.

When setting the SSU to slave mode to perform continuous reception, read SSRDR before starting the next receive operation. If the next receive operation starts before SSRDR is read and RDRF is cleared to 0, and SSRDR is read before reception completes, CE in SSSR is set to 1 after the completion of reception.

In addition, if the next receive operation starts before SSRDR is read and RDRF is cleared to 0, and SSRDR is not read until after reception completes, the receive data is discarded even though neither CE nor ORER in SSSR is set to 1.

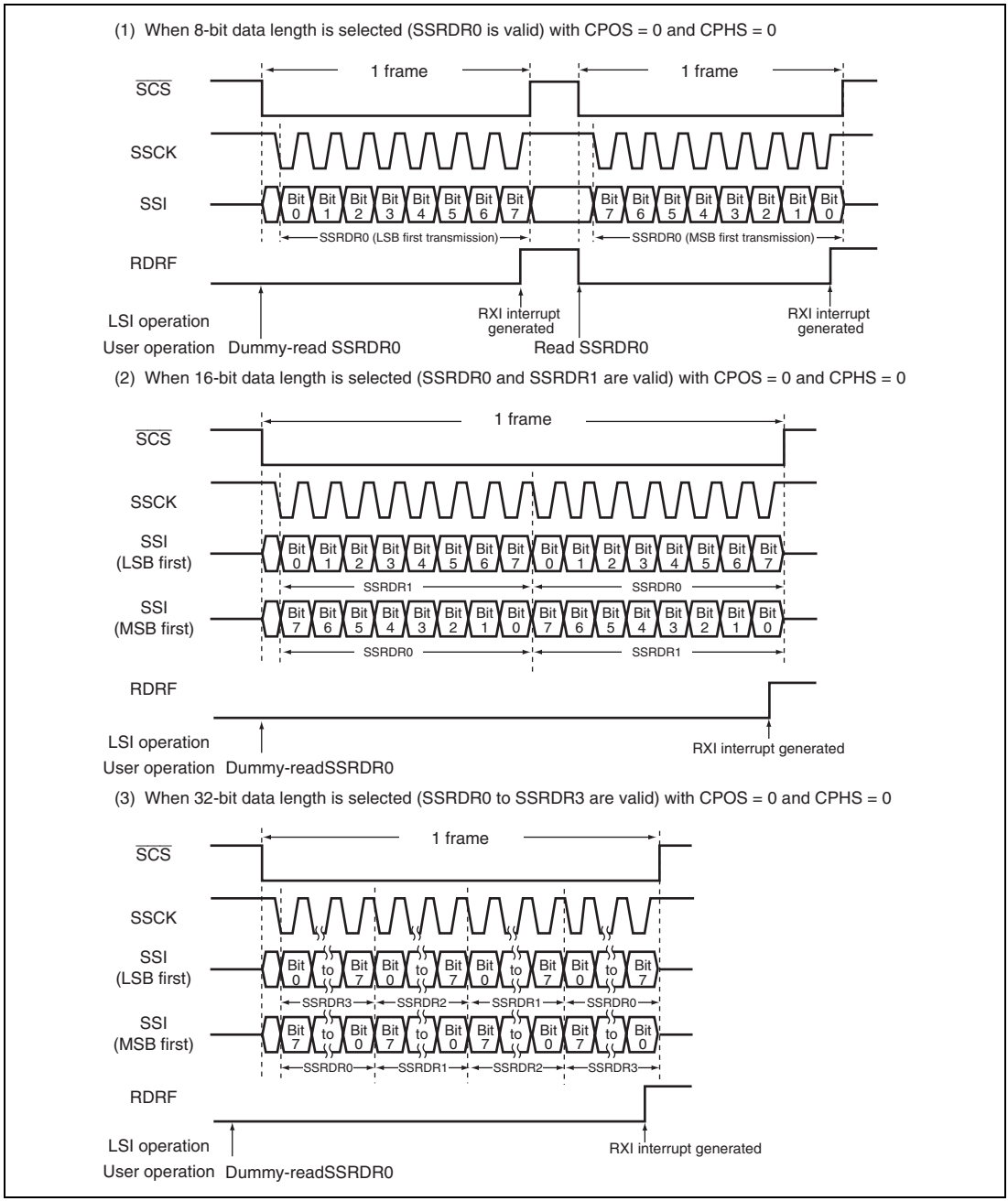
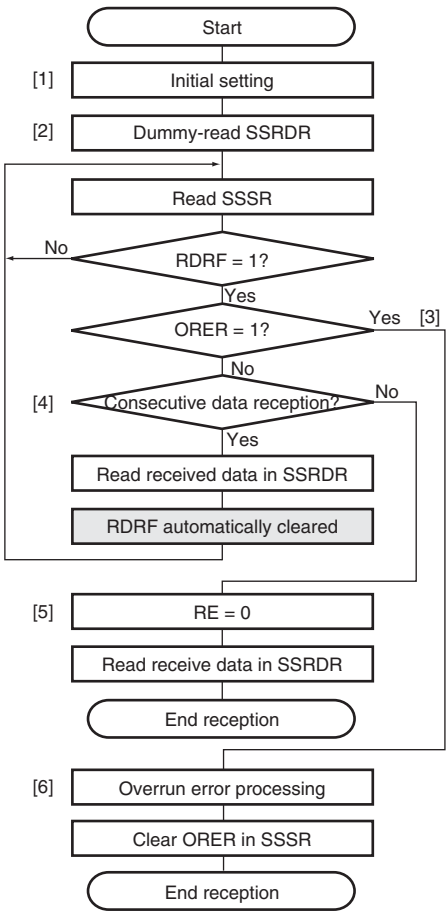



**Figure 15.7 Example of Reception Operation (SSU Mode)**




[1] Initial setting:
Specify the receive data format.

[2] Start reception:
When SSRDR is dummy-read with RE = 1, reception is started.

[3], [6] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[4] To continue single reception:
When continuing single reception, wait for time of $t_{SUcyc}$ while the RDRF flag is set to 1 and then read receive data in SSRDR.
The next single reception starts after reading receive data in SSRDR.

[5] To complete reception:
To complete reception, read receive data after clearing the RE bit to 0. When reading SSRDR without clearing the RE bit, reception is resumed.

Note: Hatching boxes represent SSU internal operations.

**Figure 15.8 Flowchart Example of Data Reception (SSU Mode)**

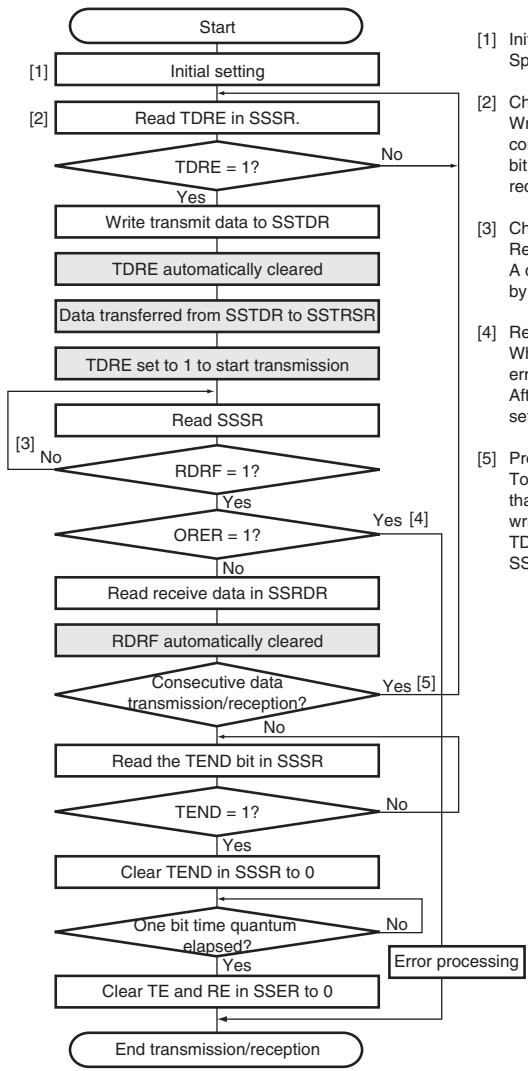
### (4) Data Transmission/Reception

Figure 15.9 shows a flowchart example of simultaneous transmission/reception. The data transmission/reception is performed combining the data transmission and data reception as mentioned above. The data transmission/reception is started by writing transmit data to SSTDR with TE = RE = 1.

Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the

transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bit to 1.

If the value of RDRF is 1 when the 8th clock rises, ORER in SSSR is set to 1, an overrun error occurs, and reception halts. Receive operation is not possible while ORER is set to 1. To restart reception, first clear ORER to 0.




[1] Initial setting:
Specify the transmit/receive data format.

[2] Check the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit in SSSR is 1. The TDRE bit is automatically cleared to 0 and transmission/reception is started by writing data to SSTDR.

[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1.
A change of the RDRF bit (from 0 to 1) can be notified by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/reception:
To continue serial data transmission/reception, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

Note: Hatching boxes represent SSU internal operations.

**Figure 15.9 Flowchart Example of Simultaneous Transmission/Reception (SSU Mode)**

### 15.4.6 $\overline{\text{SCS}}$ Pin Control and Conflict Error

When bits CSS1 and CSS0 in SSCRH are set to B'10 and the SSUMS bit in SSCRL is cleared to 0, the SCS pin becomes an input pin (Hi-Z) before the serial transfer is started and after the serial transfer is complete. Because of this, the SSU performs conflict error detection during these periods. If a low level signal is input to the SCS pin during these periods, it is detected as a conflict error. At this time, the CE bit in SSSR is set to 1 and the MSS bit is cleared to 0.

Note: While the CE bit is set to 1, transmission or reception is not resumed. Clear the CE bit to 0 before resuming the transmission or reception.

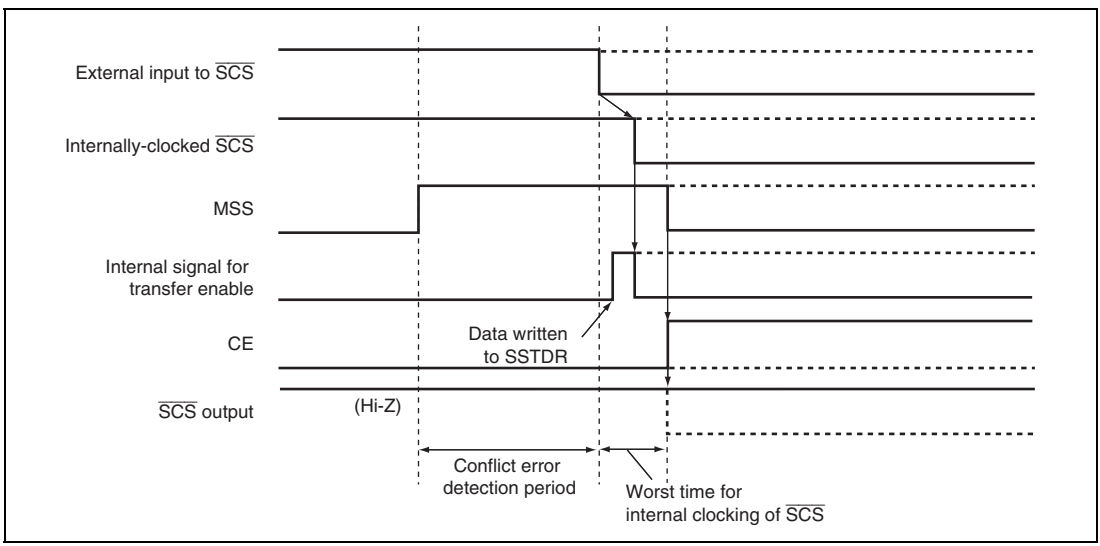



**Figure 15.10 Conflict Error Detection Timing (Before Transfer)**

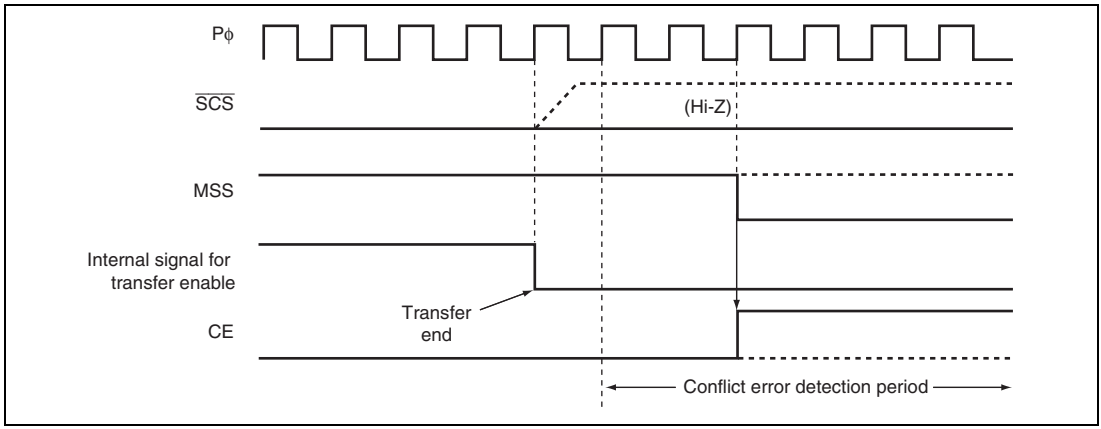



**Figure 15.11 Conflict Error Detection Timing (After Transfer End)**

### 15.4.7 Clock Synchronous Communication Mode

In clock synchronous communication mode, data communications are performed via three lines: clock line (SSCK), data input line (SSI), and data output line (SSO).

#### (1) Initial Settings in Clock Synchronous Communication Mode

Figure 15.12 shows an example of the initial settings in clock synchronous communication mode. Before data transfer, clear both the TE and RE bits in SSER to 0 to set the initial values.

Note: Before changing operating modes and communications formats, clear both the TE and RE bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the RE bit to 0 does not change the values of the RDRF and ORER bits and SSRDR. Those bits retain the previous values.

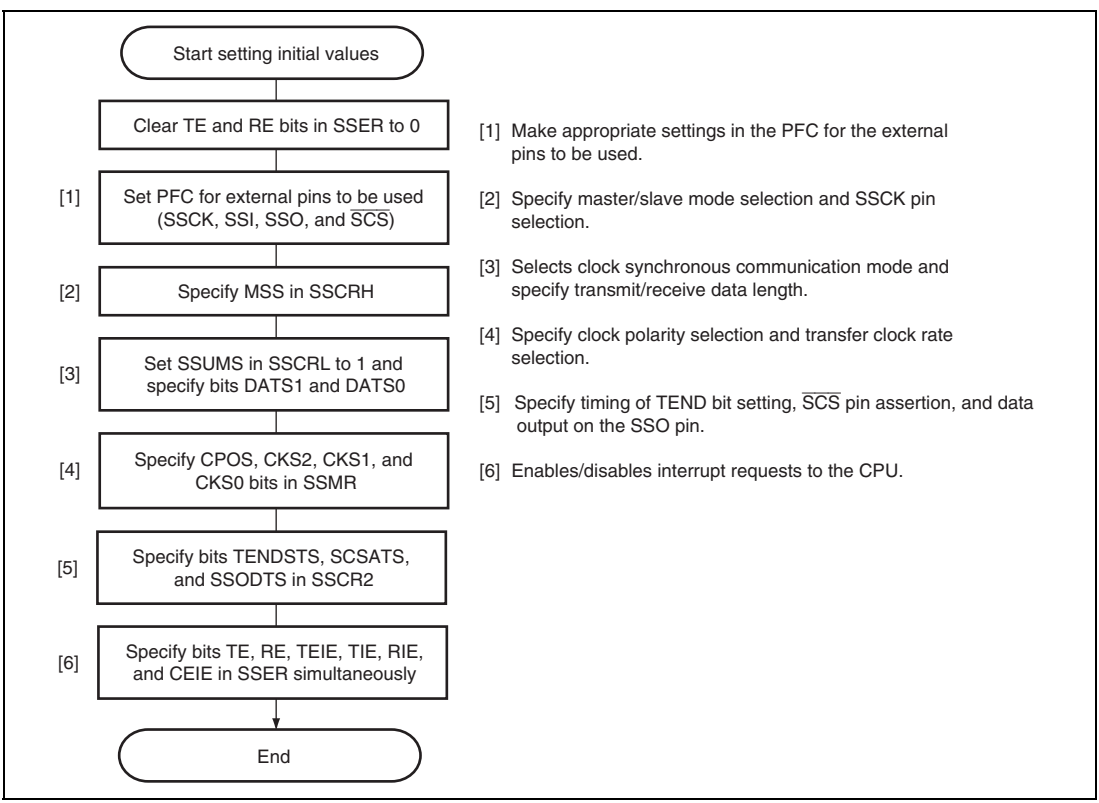



**Figure 15.12 Example of Initial Settings in Clock Synchronous Communication Mode**

### (2) Data Transmission

Figure 15.13 shows an example of transmission operation, and figure 15.14 shows a flowchart example of data transmission. When transmitting data in clock synchronous communication mode, the SSU operates as shown below.

In master mode, the SSU outputs a transfer clock and data. In slave mode, when a transfer clock is input to the SSCK pin, the SSU outputs data in synchronization with the transfer clock.

Writing transmit data to SSTDR after the TE bit is set to 1 clears the TDRE bit in SSSR to 0, and the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit to 1 and starts transmission. At this time, if the TIE bit in SSER is set to 1, a TXI interrupt is generated.

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transferred to SSTRSR to start the next frame transmission. When the 8th bit of transmit data has been transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. At this time, if the TEIE bit is set to 1, a TEI interrupt is generated.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the ORER bit is cleared to 0 before transmission.

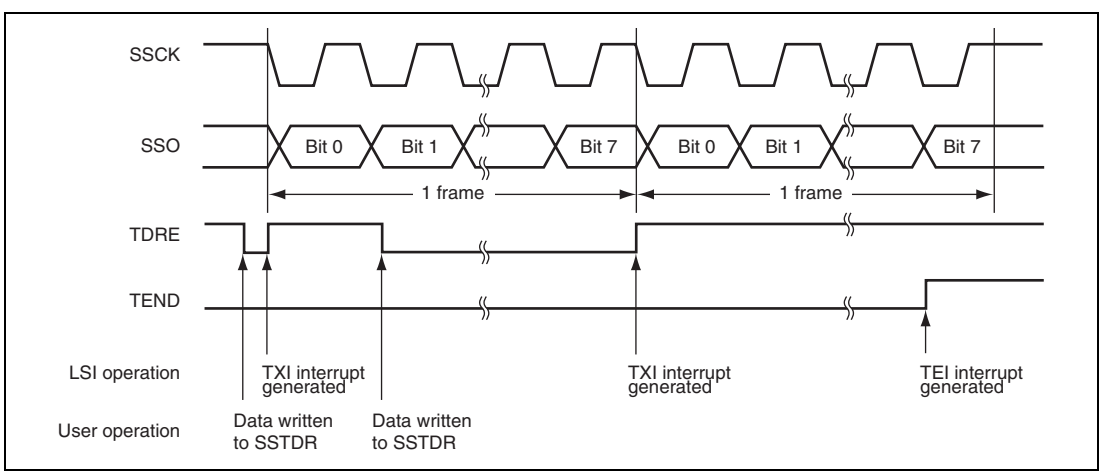
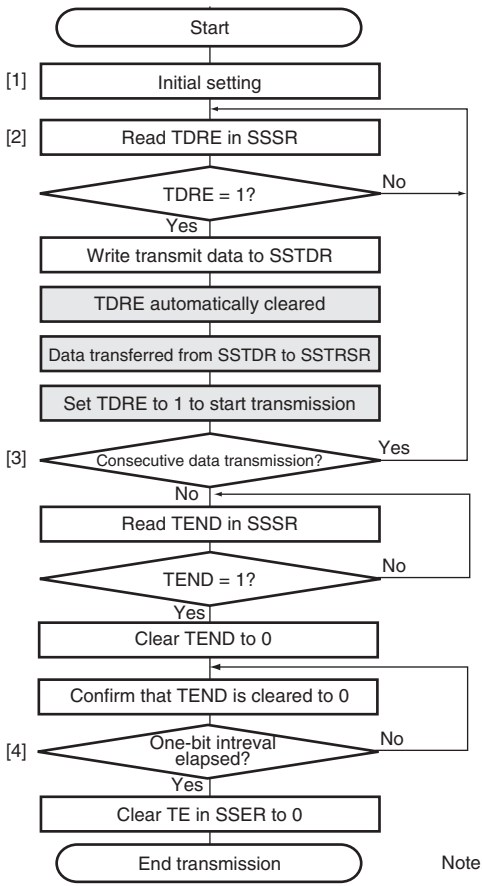



**Figure 15.13 Example of Transmission Operation (Clock Synchronous Communication Mode)**




[1] Initial setting:
Specify the transmit data format.

[2] Check that the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Procedure for consecutive data transmission:
To continue data transmission, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

[4] Procedure for data transmission end:
To end data transmission, confirm that the TEND bit is cleared to 0. After completion of transmitting the last bit, clear the TE bit to 0.

Note: Hatched boxes represent SSU internal operations.

**Figure 15.14 Flowchart Example of Transmission Operation (Clock Synchronous Communication Mode)**

### (3) Data Reception

Figure 15.15 shows an example of reception operation, and figure 15.16 shows a flowchart example of data reception. When receiving data, the SSU operates as shown below.

After setting the RE bit in SSER to 1, the SSU starts data reception.

In master mode, the SSU outputs a transfer clock and receives data. In slave mode, when a transfer clock is input to the SSCK pin, the SSU receives data in synchronization with the transfer clock.

When 1-frame data has been received, the RDRF bit in SSSR is set to 1 and the receive data is stored in SSRDR. At this time, if the RIE bit is set to 1, an RXI interrupt is generated. The RDRF bit is automatically cleared to 0 by reading SSRDR.

When setting the SSU to slave mode to perform continuous reception, read SSRDR before starting the next receive operation. If the next receive operation starts before SSRDR is read and RDRF is cleared to 0, the integrity of subsequent data cannot be guaranteed.

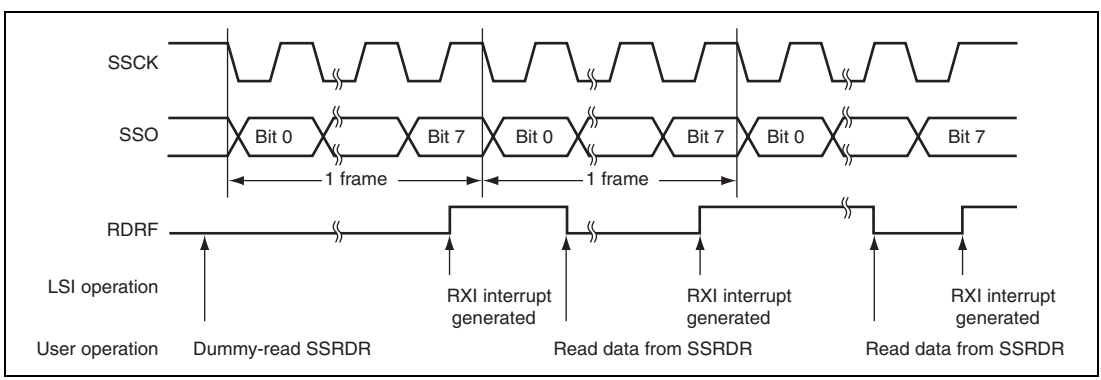
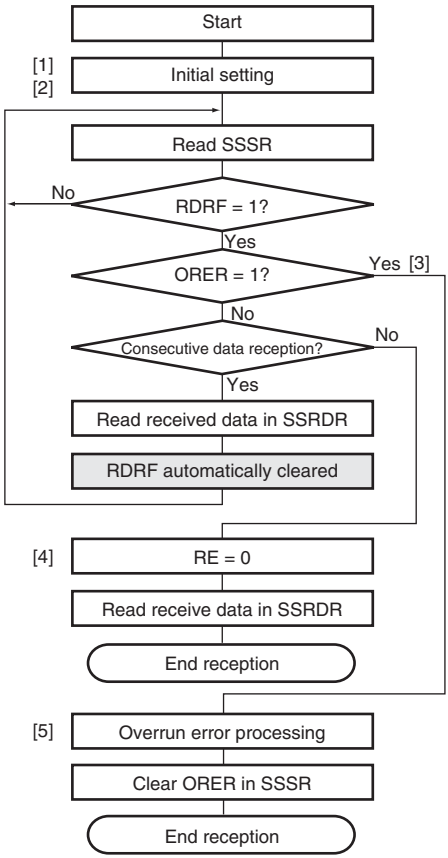



**Figure 15.15 Example of Reception Operation (Clock Synchronous Communication Mode)**




[1] Initial setting:
Specify the receive data format.

[2] Start reception:
When setting the RE bit to 1, reception is started.

[3], [5] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[4] To complete reception:
To complete reception, read receive data after clearing the RE bit to 0. When reading SSRDR without clearing the RE bit, reception is resumed.

Note: Hatching boxes represent SSU internal operations.

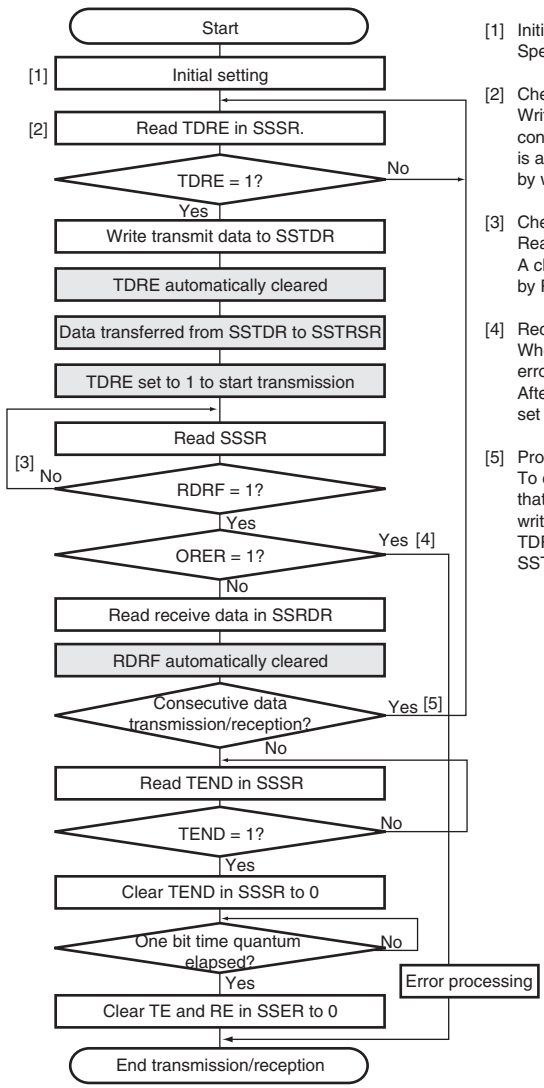
**Figure 15.16 Flowchart Example of Data Reception (Clock Synchronous Communication Mode)**

### (4) Data Transmission/Reception

Figure 15.17 shows a flowchart example of simultaneous transmission/reception. The data transmission/reception is performed combining the data transmission and data reception as mentioned above. The data transmission/reception is started by writing transmit data to SSTDR with TE = RE = 1.

Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bits to 1.

If the value of RDRF is 1 when the 8th clock rises, ORER in SSSR is set to 1, an overrun error occurs, and reception halts. Receive operation is not possible while ORER is set to 1. To restart reception, first clear ORER to 0.




[1] Initial setting:
Specify the transmit/receive data format.

[2] Check the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit in SSSR is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1.
A change of the RDRF bit (from 0 to 1) can be notified by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/reception:
To continue serial data transmission/reception, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

Note: Hatching boxes represent SSU internal operations.

**Figure 15.17 Flowchart Example of Simultaneous Transmission/Reception (Clock Synchronous Communication Mode)**

## 15.5 SSU Interrupt Sources and DTC

The SSU interrupt requests are an overrun error, a conflict error, a receive data register full, transmit data register empty, and a transmit end interrupts. Of these interrupt sources, a receive data register full, and a transmit data register empty can activate the DTC for data transfer.

Since both an overrun error and a conflict error interrupts are allocated to the SSERI vector address, and both a transmit data register empty and a transmit end interrupts are allocated to the SSTXI vector address, the interrupt source should be decided by their flags. Table 15.8 lists the interrupt sources.

When an interrupt condition shown in table 15.8 is satisfied, an interrupt is requested. Clear the interrupt source by CPU or DTC data transfer.

**Table 15.8 SSU Interrupt Sources**

| Abbreviation | Interrupt Source | Symbol | Interrupt Condition | DTC Activation |
|---|---|---|---|---|
| SSERI | Overrun error | SSOEI | (RIE = 1) • (ORER = 1) | — |
| | Conflict error | SSCEI | (CEIE = 1) • (CE = 1) | — |
| SSRXI | Receive data register full | SSRXI | (RIE = 1) • (RDRF = 1) | Yes |
| SSTXI | Transmit data register empty | SSTXI | (TIE = 1) • (TDRE = 1) | Yes |
| | Transmit end | SSTEI | (TEIE = 1) • (TEND = 1) | — |

## 15.6 Usage Notes

### 15.6.1 Module Standby Mode Setting

The SSU operation can be disabled or enabled using the standby control register. The initial setting is for SSU operation to be halted. Access to registers is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes

### 15.6.2 Access to SSTDR and SSRDR Registers

Do not access SSTDR and SSRDR registers not validated by the setting of the DATS bits of the SSCRL register. If accessed, transmission or reception thereafter may not be performed normally.

### 15.6.3 Continuous Transmission/Reception in SSU Slave Mode

During continuous transmission/reception in SSU slave mode, negate the $\overline{\text{SCS}}$ pin (high level) for every frame. If the $\overline{\text{SCS}}$ pin is kept asserted (low level) for more than one frame, transmission or reception cannot be performed correctly.

### 15.6.4 Note for Reception Operations in SSU Slave Mode

In continuous reception when slave reception in SSU mode has been selected, read the SS receive-data register (SSRDR) before each next round of reception starts (i.e. before an externally connected master device starts a next round of transmission).

If the next round of reception starts after the SS status register receive-data full (RDRF) bit has been set to 1 but before the SSRDR has been read, and the SSRDR is read before the reception of one frame is complete, the conflict/incomplete error bit in SSSR will be set to 1 on completion of reception.

Furthermore, when the next round of reception starts after the receive-data full (RDRF) bit has been set to 1 and before the SSRDR has been read, and the SSRDR has not been read by the end of the reception of the frame, the CE and overflow-error (ORER) bits will not have been set, but the received data will be discarded.

Further note that this point for caution does not apply to simultaneous transmission and reception in SSU slave mode or to clock-synchronous mode.

### 15.6.5 Note on Master Transmission and Master Reception Operations in SSU Mode

To perform master transmission or reception in SSU mode, perform one of the following operations:

- After the TDRE flag in the SSSR register is set to 1, store the next byte of transmit data in SSTDR before transmission of the second to last bit starts.
- Store the next byte of transmit data in SSTDR after confirming that the TEND flag in the SSSR register has been set to 1.
- Use the SSU with TENDSTS in the SSCR2 register cleared to 0, or with both TENDSTS and SCSATS in the SSCR2 register set to 1.

### 15.6.6 Note on DTC Transfers

When a DTC transfer occurs with SSTXI as the activation source, TDRE is not cleared when the transfer counter reaches H'0000 but communication operation starts anyway.

When using the SSTXI interrupt to clear the flag, perform interrupt handling first.

However, do not clear the flag within the SSTXI interrupt handler when the initial value of the DTC's transfer counter is set to H'0001 and DISEL is set to 1. In this case, clearing the flag by the interrupt handler may cause the SSU to start communication operation a second time.

# Section 16 I²C Bus Interface 2 (I²C2)

The I²C bus interface 2 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.

Figure 16.1 shows a block diagram of the I²C bus interface 2. Figure 16.2 shows an example of I/O pin connections to external circuits.

## 16.1 Features

- Selection of I²C format or clock 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.
- Module standby mode can be set

**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 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 data transfer controller (DTC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.

**Clock synchronous serial format:**

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error

  The data transfer controller (DTC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.

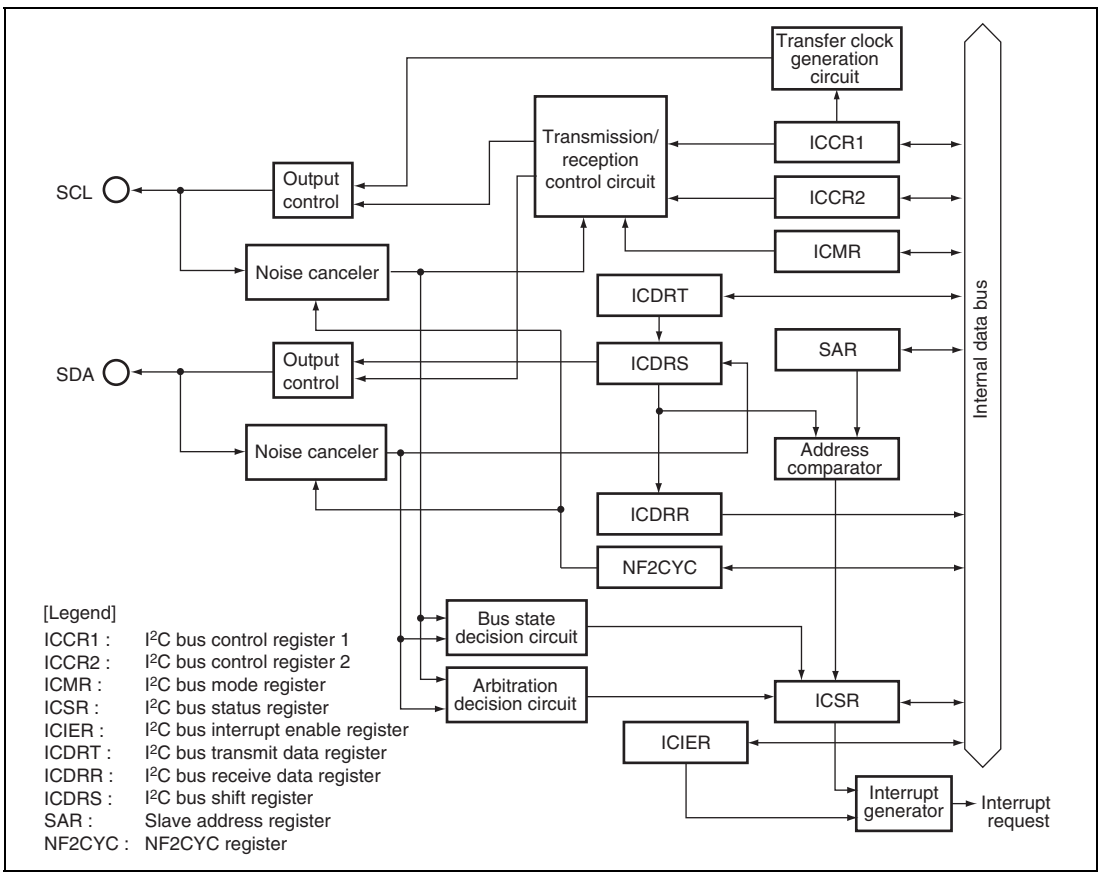



**Figure 16.1 Block Diagram of I²C Bus Interface 2**

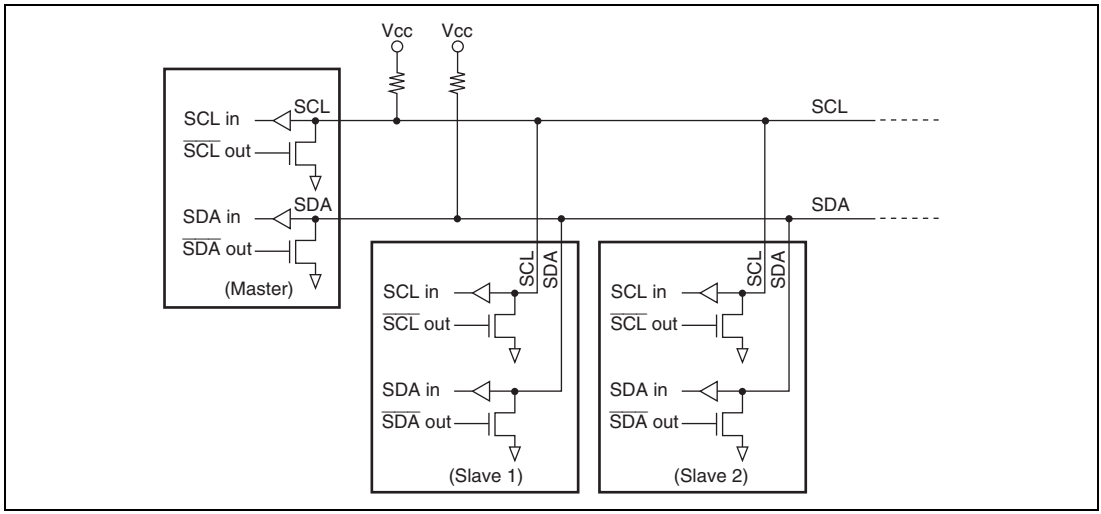



**Figure 16.2 External Circuit Connections of I/O Pins**

## 16.2 Input/Output Pins

Table 16.1 shows the pin configuration for the I²C bus interface 2.

**Table 16.1 I²C Bus Interface Pin Configuration**

| Pin Name | Symbol | I/O | Function |
|---|---|---|---|
| Serial clock | SCL | I/O | I²C serial clock input/output |
| Serial data | SDA | I/O | I²C serial data input/output |

## 16.3 Register Descriptions

The I²C bus interface 2 has the following registers. For details on register addresses and register states during each processing, refer to section 25, List of Registers.

**Table 16.2 Register Configuration**

| Register Name | Abbreviation | R/W | Initial value | Address | Access Size |
|---|---|---|---|---|---|
| I²C bus control register 1 | ICCR1 | R/W | H'00 | H'FFFFCD80 | 8 |
| I²C bus control register 2 | ICCR2 | R/W | H'7D | H'FFFFCD81 | 8 |
| I²C bus mode register | ICMR | R/W | H'38 | H'FFFFCD82 | 8 |
| I²C bus interrupt enable register | ICIER | R/W | H'00 | H'FFFFCD83 | 8 |
| I²C bus status register | ICSR | R/W | H'00 | H'FFFFCD84 | 8 |
| I²C bus slave address register | SAR | R/W | H'00 | H'FFFFCD85 | 8 |
| I²C bus transmit data register | ICDRT | R/W | H'FF | H'FFFFCD86 | 8 |
| I²C bus receive data register | ICDRR | R/W | H'FF | H'FFFFCD87 | 8 |
| NF2CYC register | NF2CYC | R/W | H'00 | H'FFFFCD88 | 8 |

### 16.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 is an 8-bit readable/writable register that enables or disables the I²C bus interface 2, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master 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 2 Enable<br>0: This module is halted.<br>1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | RCVD | 0 | R/W | Reception Disable<br>When TRS = 0, this bit enables or disables continuous reception without reading of ICDRR. In master receive mode, when ICDRR cannot be read before the rising edge of the 8th clock of SCL, set RCVD to 1 so that data is received in byte units.<br>0: Enables continuous reception<br>1: Disables continuous reception |
| 5 | MST | 0 | R/W | Master/Slave Select |
| 4 | TRS | 0 | R/W | Transmit/Receive Select<br>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.<br>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 clock synchronous serial format, MST is cleared and the mode changes to slave receive mode.<br>Operating modes are described below according to MST and TRS combination. When clock synchronous serial format is selected and MST = 1, clock is output.<br>00: Slave receive mode<br>01: Slave transmit mode<br>10: Master receive mode<br>11: Master transmit mode |
| 3 to 0 | CKS[3:0] | 0000 | R/W | Transfer Clock Select 3 to 0<br>These bits should be set according to the necessary transfer rate (table 16.3) in master mode. In slave mode, these bits should be used to specify the data setup time in transmission mode. The setup time is set to 10 tpcyc when CKS3 = 0 or 20 tpcyc when CKS3 = 1 (tpcyc is one Pϕ cycle). |

## Table 16.3 Transfer Rate

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | | Transfer Rate | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| CKS3 | CKS2 | CKS1 | CKS0 | Clock | Pϕ=10 MHz | Pϕ=16 MHz | Pϕ=20 MHz | Pϕ=25 MHz | Pϕ=33 MHz | Pϕ=40 MHz |
| 0 | 0 | 0 | 0 | Pϕ/28 | 357 kHz | 571 kHz | 714 kHz | 893 kHz | 1.18 MHz | 1.43 MHz |
| | | | 1 | Pϕ/40 | 250 kHz | 400 kHz | 500 kHz | 625 kHz | 825 kHz | 1.00 MHz |
| | | 1 | 0 | Pϕ/48 | 208 kHz | 333 kHz | 417 kHz | 521 kHz | 688 kHz | 833 kHz |
| | | | 1 | Pϕ/64 | 156 kHz | 250 kHz | 313 kHz | 391 kHz | 516 kHz | 625 kHz |
| | 1 | 0 | 0 | Pϕ/80 | 125 kHz | 200 kHz | 250 kHz | 313 kHz | 413 kHz | 500 kHz |
| | | | 1 | Pϕ/100 | 100 kHz | 160 kHz | 200 kHz | 250 kHz | 330 kHz | 400 kHz |
| | | 1 | 0 | Pϕ/112 | 89.3 kHz | 143 kHz | 179 kHz | 223 kHz | 295 kHz | 357 kHz |
| | | | 1 | Pϕ/128 | 78.1 kHz | 125 kHz | 156 kHz | 195 kHz | 258 kHz | 313 kHz |
| 1 | 0 | 0 | 0 | Pϕ/112 | 89.3 kHz | 143 kHz | 179 kHz | 223 kHz | 295 kHz | 357 kHz |
| | | | 1 | Pϕ/160 | 62.5 kHz | 100 kHz | 125 kHz | 156 kHz | 206 kHz | 250 kHz |
| | | 1 | 0 | Pϕ/192 | 52.1 kHz | 83.3 kHz | 104 kHz | 130 kHz | 172 kHz | 208 kHz |
| | | | 1 | Pϕ/256 | 39.1 kHz | 62.5 kHz | 78.1 kHz | 97.7 kHz | 129 kHz | 156 kHz |
| | 1 | 0 | 0 | Pϕ/320 | 31.3 kHz | 50.0 kHz | 62.5 kHz | 78.1 kHz | 103 kHz | 125 kHz |
| | | | 1 | Pϕ/400 | 25.0 kHz | 40.0 kHz | 50.0 kHz | 62.5 kHz | 82.5 kHz | 100 kHz |
| | | 1 | 0 | Pϕ/448 | 22.3 kHz | 35.7 kHz | 44.6 kHz | 55.8 kHz | 73.7 kHz | 89.3 kHz |
| | | | 1 | Pϕ/512 | 19.5 kHz | 31.3 kHz | 39.1 kHz | 48.8 kHz | 64.5 kHz | 78.1 kHz |

### 16.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR2 is an 8-bit readable/writable register that issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus interface 2.

| 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<br>This bit enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the clock 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. To issue a start condition, simultaneously write 1 to BBSY and 0 to SCP. Follow this procedure also when transmitting a repeated start condition. To issue a stop condition, simultaneously write 0 to BBSY and 0 to SCP. |
| 6 | SCP | 1 | R/W | Start/Stop Issue Condition Disable<br>The SCP bit controls the issue of start/stop conditions in master mode.<br>To issue a start condition, simultaneously write 1 to BBSY and 0 to SCP. A repeated start condition is issued in the same way. To issue a stop condition, simultaneously write 0 to BBSY and 0 to 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<br>This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer.<br>0: When reading, SDA pin outputs low.<br>When writing, SDA pin is changed to output low.<br>1: When reading, SDA pin outputs high.<br>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<br>This bit 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 | This bit monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low. |
| 2 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 1 | IICRST | 0 | R/W | IIC Control Part Reset<br>This bit resets the control part except for I²C registers. If this bit is set to 1 when hang-up occurs because of communication failure during I²C operation, some of I²C registers and control part can be reset. |
| 0 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |

### 16.3.3 I²C Bus Mode Register (ICMR)

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first and selects the transfer bit count.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MLS | - | - | - | BCWP | BC[2:0] | | |
| Initial value: | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| R/W: | R/W | R | R | R | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | MLS | 0 | R/W | MSB-First/LSB-First Select<br>0: MSB-first<br>1: LSB-first<br>Set this bit to 0 when the I²C bus format is used. |
| 6 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 5, 4 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. |
| 3 | BCWP | 1 | R/W | BC Write Protect<br>This bit controls the BC2 to BC0 modifications. When modifying BC2 to BC0, this bit should be cleared to 0. In clock synchronous serial mode, BC should not be modified.<br>0: When writing, values of BC2 to BC0 are set.<br>1: When reading, 1 is always read.<br>When writing, settings of BC2 to BC0 are invalid. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 to 0 | BC[2:0] | 000 | R/W | Bit Counter 2 to 0<br>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 bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL pin is low. The value returns to 000 at the end of a data transfer, including the acknowledge bit. These bits are automatically set to 111 after a stop condition is detected. These bits are cleared by a power-on reset and in standby mode. These bits are also cleared by setting IICRST of ICCR2 to 1. With the clock synchronous serial format, these bits should not be modified.<br>I²C Bus Format / Clock Synchronous Serial Format<br>000: 9 bits / 000: 8 bits<br>001: 2 bits / 001: 1 bit<br>010: 3 bits / 010: 2 bits<br>011: 4 bits / 011: 3 bits<br>100: 5 bits / 100: 4 bits<br>101: 6 bits / 101: 5 bits<br>110: 7 bits / 110: 6 bits<br>111: 8 bits / 111: 7 bits |

### 16.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER is an 8-bit readable/writable register that enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits received.

| 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<br>When the TDRE bit in ICSR is set to 1 or 0, this bit enables or disables the transmit data empty interrupt (IITXI).<br>0: Transmit data empty interrupt request (IITXI) is disabled.<br>1: Transmit data empty interrupt request (IITXI) is enabled. |
| 6 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br>This bit enables or disables the transmit end interrupt (IITEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. IITEI can be canceled by clearing the TEND bit or the TEIE bit to 0.<br>0: Transmit end interrupt request (IITEI) is disabled.<br>1: Transmit end interrupt request (IITEI) is enabled. |
| 5 | RIE | 0 | R/W | Receive Interrupt Enable<br>RIE enables or disables the receive data full interrupt request (IIRXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. IIRXI can be canceled by clearing the RDRF or RIE bit to 0.<br>0: Receive data full interrupt request (IIRXI) are disabled.<br>1: Receive data full interrupt request (IIRXI) are enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | NAKIE | 0 | R/W | NACK Receive Interrupt Enable<br>NAKIE enables or disables the NACK detection/arbitration lost/overrun error interrupt request (IINAKI) when the NACKF or AL/OVE bit in ICSR is set to 1. IINAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0.<br>0: NACK receive interrupt request (IINAKI) is disabled.<br>1: NACK receive interrupt request (IINAKI) is enabled. |
| 3 | STIE | 0 | R/W | Stop Condition Detection Interrupt Enable<br>This bit enables or disables the stop condition detection interrupt request (IISTPI) when the STOP bit in ICSR is set.<br>0: Stop condition detection interrupt request (IISTPI) is disabled.<br>1: Stop condition detection interrupt request (IISTPI) is enabled. |
| 2 | ACKE | 0 | R/W | Acknowledge Bit Judgment Select<br>0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed.<br>1: If the receive acknowledge bit is 1, continuous transfer is halted. |
| 1 | ACKBR | 0 | R | Receive Acknowledge<br>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.<br>0: Receive acknowledge = 0<br>1: Receive acknowledge = 1 |
| 0 | ACKBT | 0 | R/W | Transmit Acknowledge<br>In receive mode, this bit specifies the bit to be sent at the acknowledge timing.<br>0: 0 is sent at the acknowledge timing.<br>1: 1 is sent at the acknowledge timing. |

### 16.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that confirms interrupt request flags and their status.

| 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<br>[Setting conditions]<br>• When data is transferred from ICDRT to ICDRS and ICDRT becomes empty<br>• When TRS is set<br>• When the start condition (including retransmission) is issued<br>• When slave mode is changed from receive mode to transmit mode<br>[Clearing conditions]<br>• When 0 is written to TDRE after reading TDRE = 1<br>• When data is written to ICDRT<br>• DTC is activated by IITXI interrupt and the DISEL bit in MRB of DTC is 0. |
| 6 | TEND | 0 | R/W | Transmit End<br>[Setting conditions]<br>• When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1<br>• When the final bit of transmit frame is sent with the clock synchronous serial format<br>[Clearing conditions]<br>• When 0 is written to TEND after reading TEND = 1<br>• When data is written to ICDRT<br>• DTC is activated by IITXI interrupt and the DISEL bit in MRB of DTC is 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | RDRF | 0 | R/W | Receive Data Register Full<br>[Setting condition]<br>• When a receive data is transferred from ICDRS to ICDRR<br>[Clearing conditions]<br>• When 0 is written to RDRF after reading RDRF = 1<br>• When ICDRR is read<br>• DTC is activated by IIRXI interrupt and the DISEL bit in MRB of DTC is 0. |
| 4 | NACKF | 0 | R/W | No Acknowledge Detection Flag*<br>[Setting condition]<br>• When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1<br>[Clearing condition]<br>• When 0 is written to NACKF after reading NACKF = 1 |
| 3 | STOP | 0 | R/W | Stop Condition Detection Flag<br>[Setting conditions]<br>• In master mode, when a stop condition is detected after frame transfer<br>• In slave mode, when a stop condition is detected after the slave address in the first byte that came following the detection of a start condition have matched the address set in SAR.<br>[Clearing condition]<br>• When 0 is written to STOP after reading STOP = 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | AL/OVE | 0 | R/W | Arbitration Lost Flag/Overrun Error Flag<br>This 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 clock synchronous format.<br>When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface 2 detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been occupied by another master.<br>[Setting conditions]<br>• If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode<br>• When the SDA pin outputs high in master mode while a start condition is detected<br>• When the final bit is received with the clock synchronous format while RDRF = 1<br>[Clearing condition]<br>• When 0 is written to AL/OVE after reading AL/OVE = 1 |
| 1 | AAS | 0 | R/W | Slave Address Recognition Flag<br>In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR.<br>[Setting conditions]<br>• When the slave address is detected in slave receive mode<br>• When the general call address is detected in slave receive mode.<br>[Clearing condition]<br>• When 0 is written to AAS after reading AAS = 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | ADZ | 0 | R/W | General Call Address Recognition Flag<br>This bit is valid in slave receive mode with the I²C bus format.<br>[Setting condition]<br>• When the general call address is detected in slave receive mode<br>[Clearing condition]<br>• When 0 is written to ADZ after reading ADZ = 1 |

Note: * When NACKF = 1 is detected, be sure to clear NACKF in the transfer end processing. Until the flag is cleared, next transmission or reception cannot be started.

### 16.3.6 I²C Bus 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.

| 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] | All 0 | R/W | Slave Address 6 to 0<br>These bits set a unique address in bits SVA6 to SVA0, differing form the addresses of other slave devices connected to the I²C bus. |
| 0 | FS | 0 | R/W | Format Select<br>0: I²C bus format is selected<br>1: Clock synchronous serial format is selected |

### 16.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. If ICDRT is written to and then read while the MLS bit in ICMR is set to 1, data is read in the reversed order (MSB–LSB order is reversed). ICDRT is initialized to H'FF.

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

### 16.3.8 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.

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

### 16.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.

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

### 16.3.10 NF2CYC Register (NF2CYC)

NF2CYC is an 8-bit readable/writable register that selects the range of the noise filtering for the SCL and SDA pins. For details of the noise filter, see section 16.4.7, Noise Filter.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | NF2CYC |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | NF2CYC | 0 | R/W | Noise Filtering Range Select<br>0: The noise less than one cycle of the peripheral clock can be filtered out<br>1: The noise less than two cycles of the peripheral clock can be filtered out |

## 16.4 Operation

The I²C bus interface 2 can communicate either in I²C bus mode or clock synchronous serial mode by setting FS in SAR.

### 16.4.1 I²C Bus Format

Figure 16.3 shows the I²C bus formats. Figure 16.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

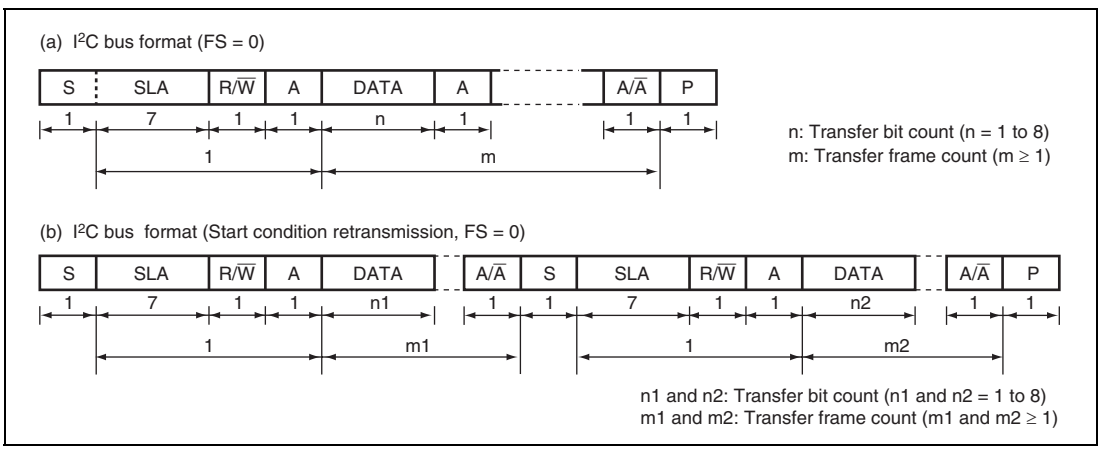



**Figure 16.3 I²C Bus Formats**

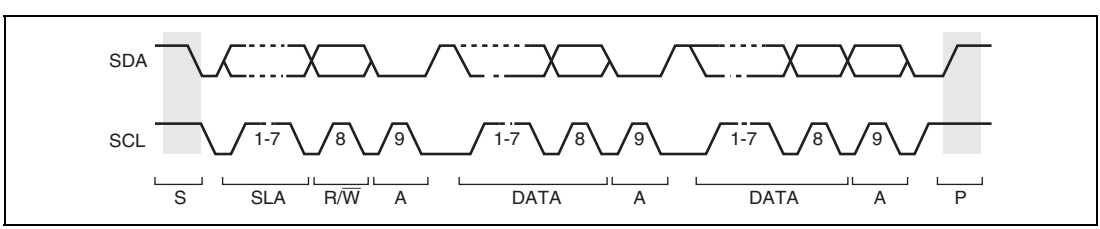



**Figure 16.4 I²C Bus Timing**

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/W: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

### 16.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 16.5 and 16.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS bit in ICMR and bits CKS3 to CKS0 in ICCR1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. 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.

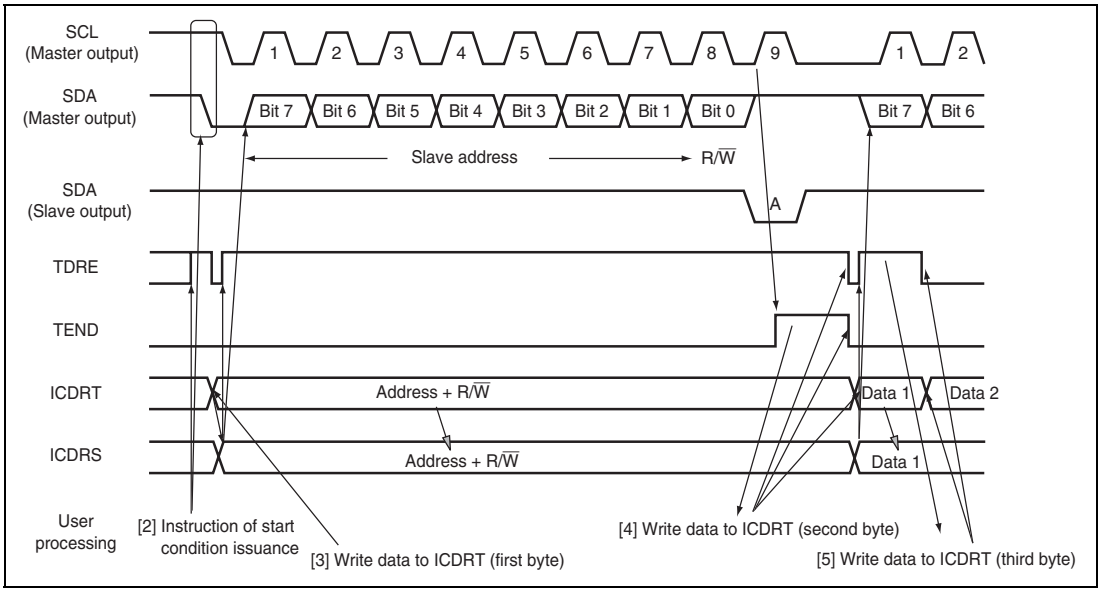



**Figure 16.5 Master Transmit Mode Operation Timing (1)**

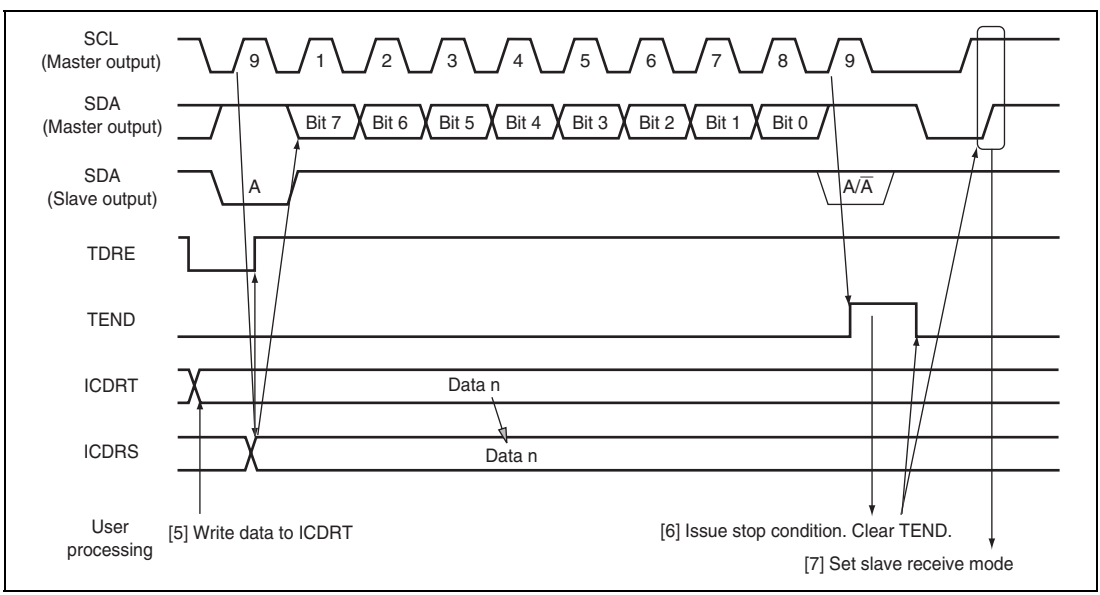



**Figure 16.6 Master Transmit Mode Operation Timing (2)**

### 16.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 16.7 and 16.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICSR is set to 1 at the rise of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

Note: If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in ICCR1 is set.

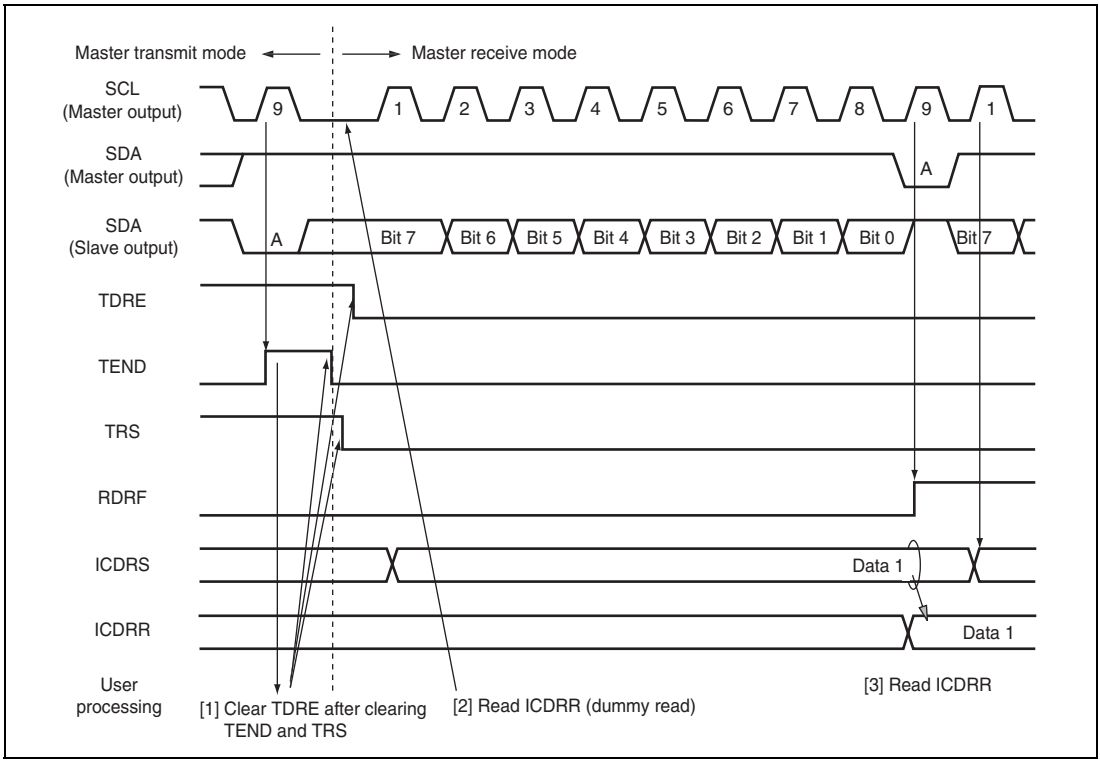



**Figure 16.7 Master Receive Mode Operation Timing (1)**

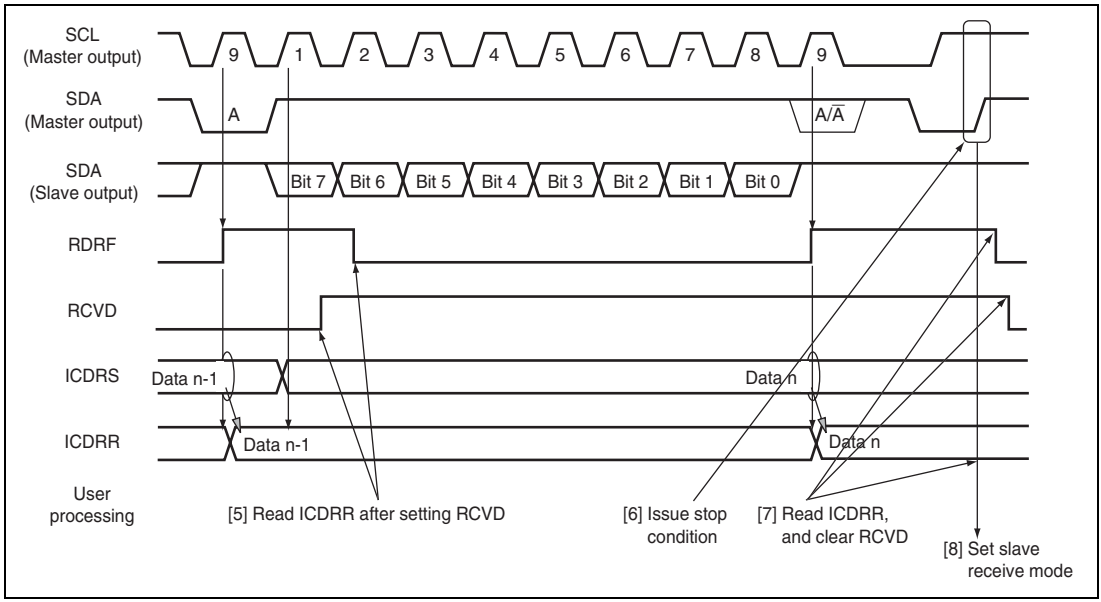



**Figure 16.8 Master Receive Mode Operation Timing (2)**

### 16.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, refer to figures 16.9 and 16.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS bit in ICMR and bits CKS3 to CKS0 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 free.
5. Clear TDRE.

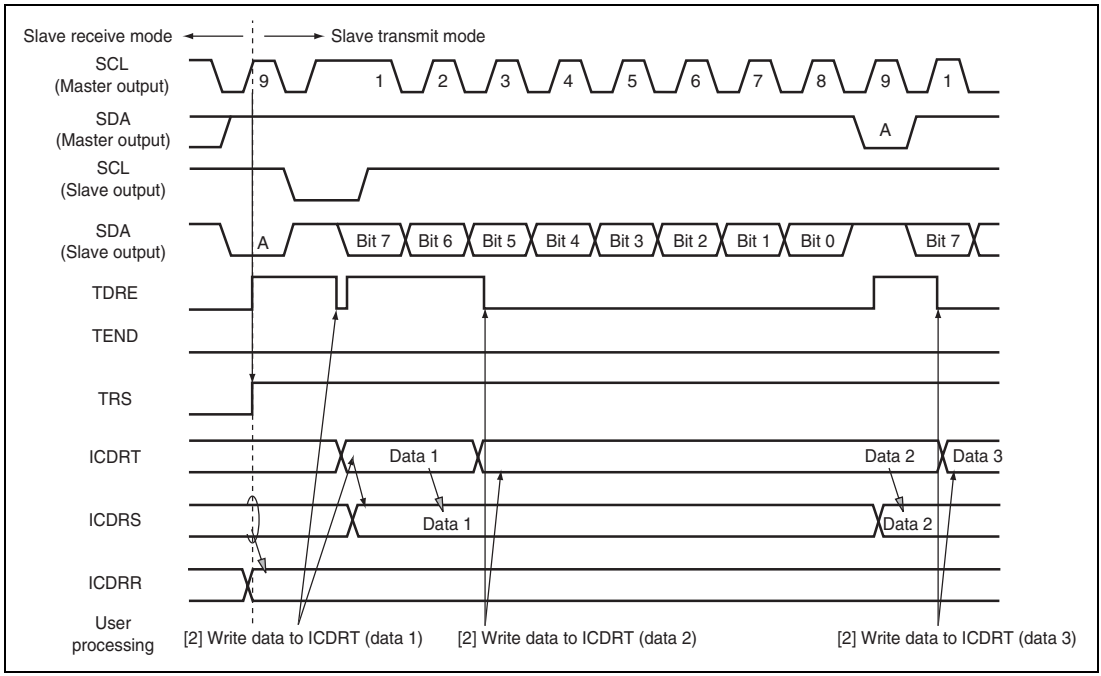



**Figure 16.9 Slave Transmit Mode Operation Timing (1)**

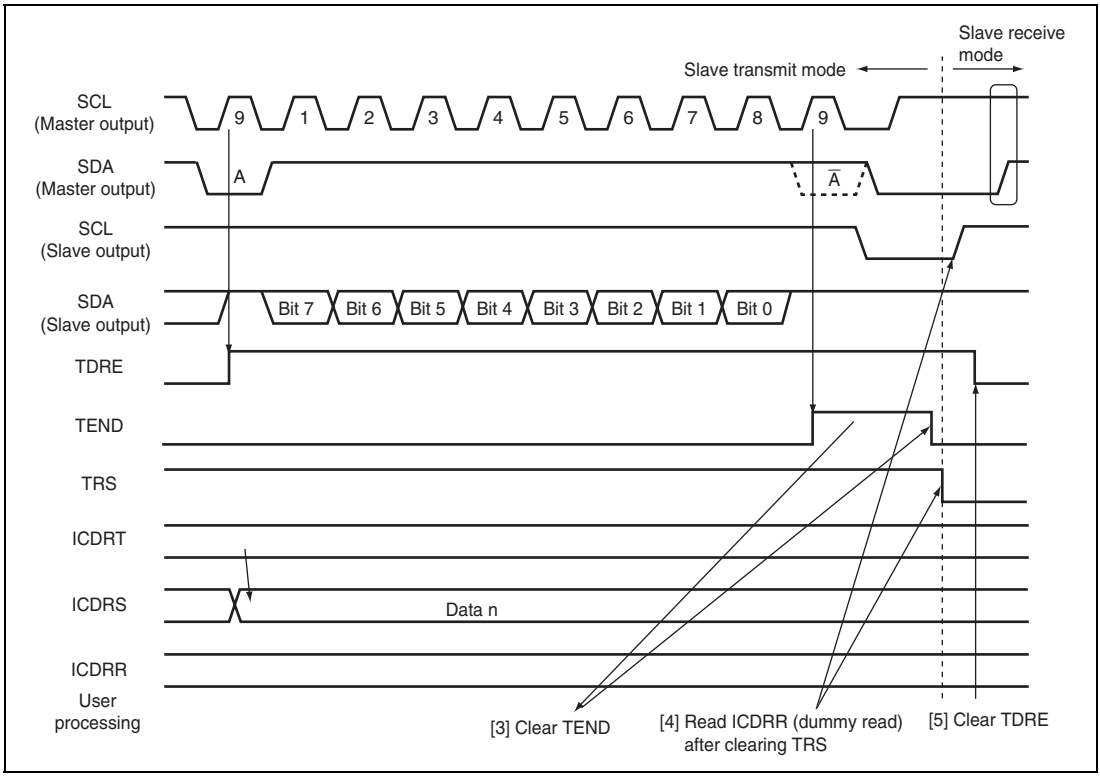



**Figure 16.10 Slave Transmit Mode Operation Timing (2)**

### 16.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, refer to figures 16.11 and 16.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS bit in ICMR and bits CKS3 to CKS0 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.

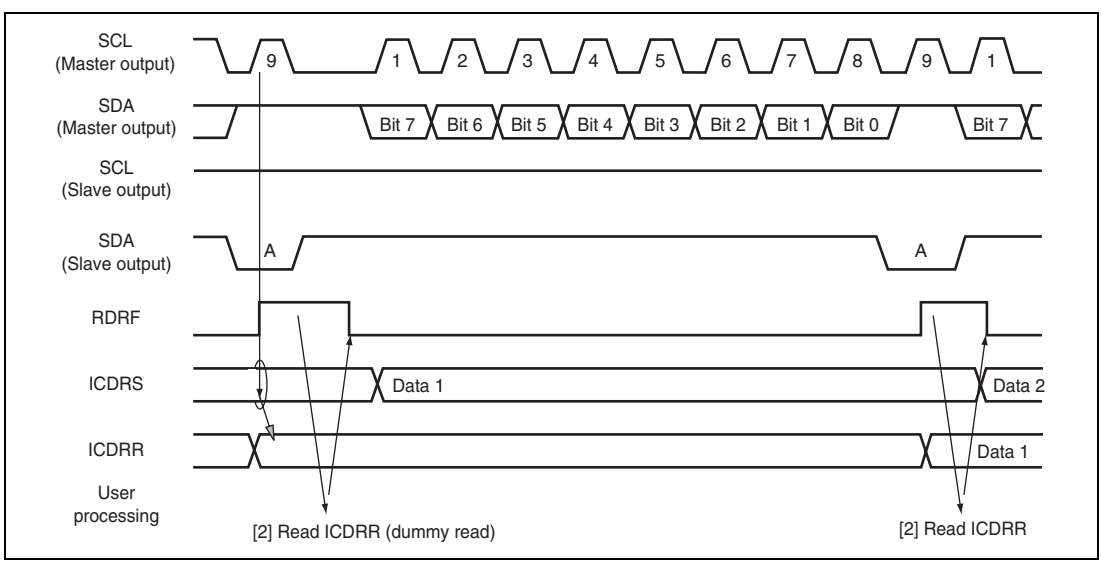



**Figure 16.11 Slave Receive Mode Operation Timing (1)**

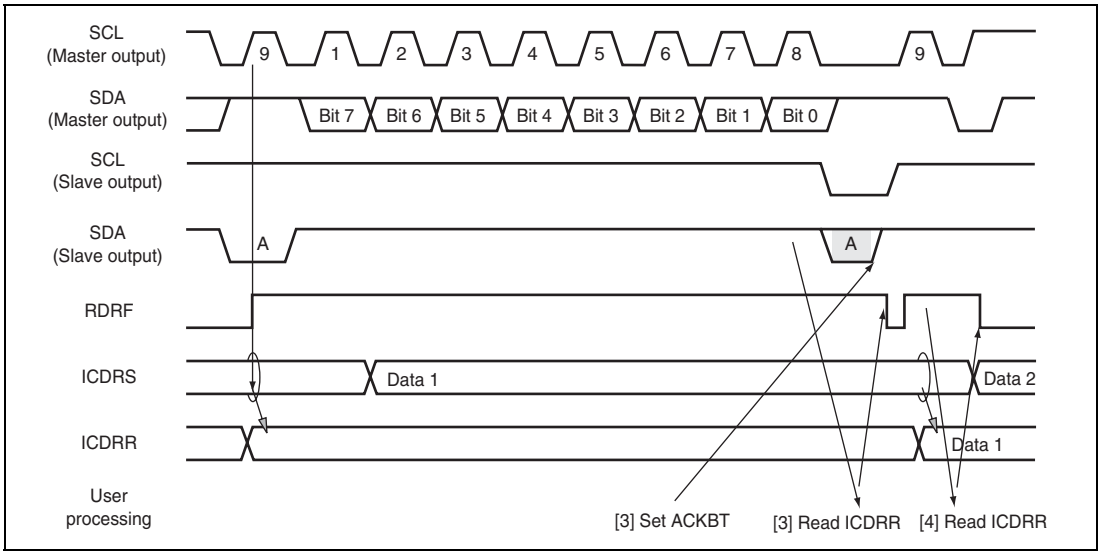



**Figure 16.12 Slave Receive Mode Operation Timing (2)**

### 16.4.6 Clock Synchronous Serial Format

This module can be operated with the clock synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

#### (1) Data Transfer Format

Figure 16.13 shows the clock synchronous serial transfer format.

The transfer data is output from the fall to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

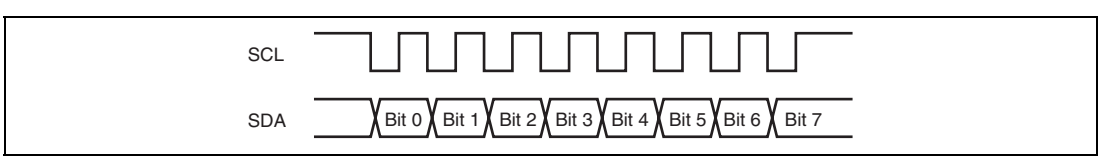



**Figure 16.13 Clock Synchronous Serial Transfer Format**

### (2) Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, refer to figure 16.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 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.

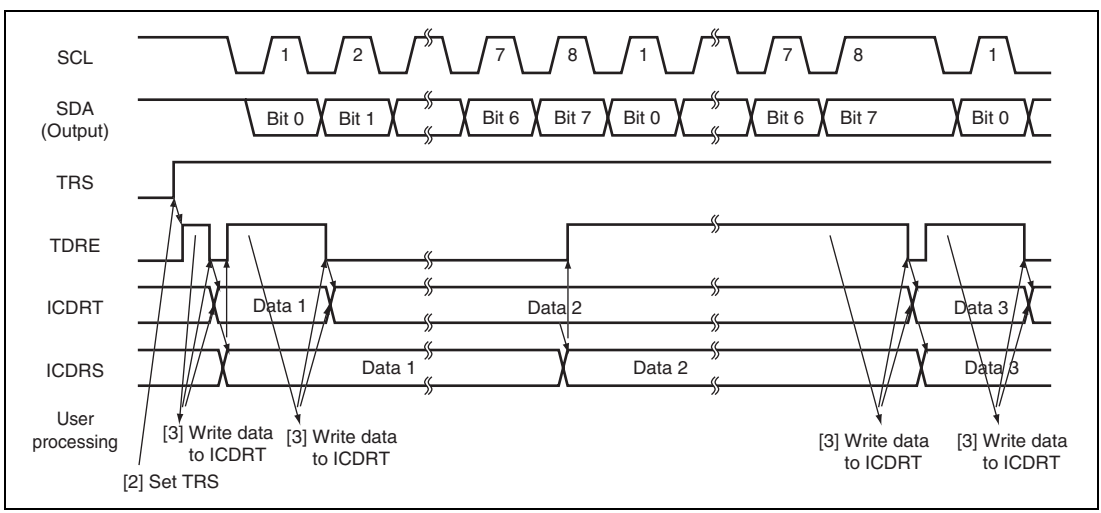



**Figure 16.14 Transmit Mode Operation Timing**

### (3) Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refer to figure 16.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

Notes: Follow the steps below to receive only one byte with MST=1 specified. See figure 16.16 for the operation timing.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS3 to CKS0 in ICCR1. (Initial setting)
2. Set MST=1 while the RCVD bit in ICCR1 is 0. This causes the receive clock to be output.
3. Check if the BC2 bit in ICMR is set to 1 and then set the RCVD bit in ICCR1 to 1. This causes the SCL to be fixed to the high level after outputting one byte of the receive clock.

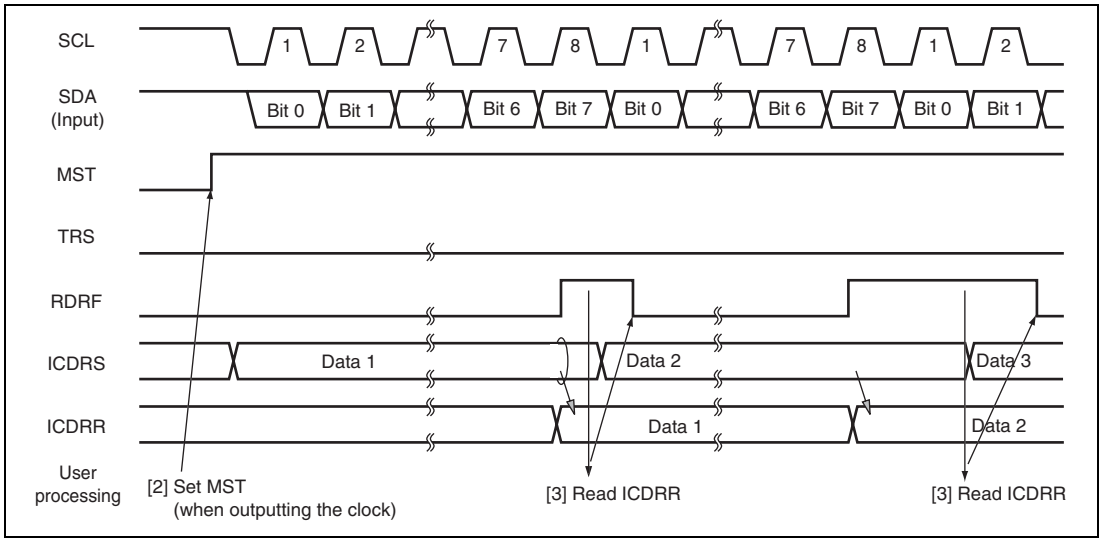



**Figure 16.15 Receive Mode Operation Timing**

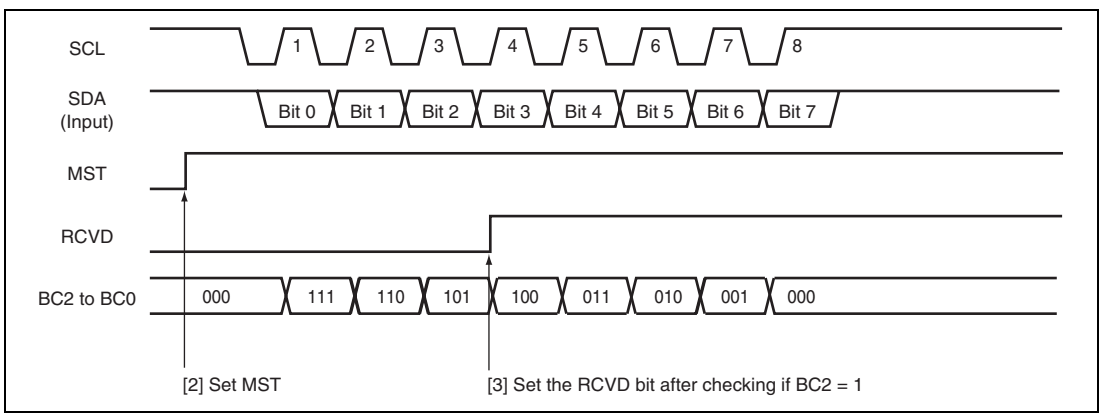



**Figure 16.16 Operation Timing For Receiving One Byte**

### 16.4.7 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 16.17 shows a block diagram of the noise filter circuit.

The noise filter consists of three cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the peripheral clock. When NF2CYC is set to 0, this signal is not passed forward to the next circuit unless the outputs of both latches agree. When NF2CYC is set to 1, this signal is not passed forward to the next circuit unless the outputs of three latches agree. If they do not agree, the previous value is held.

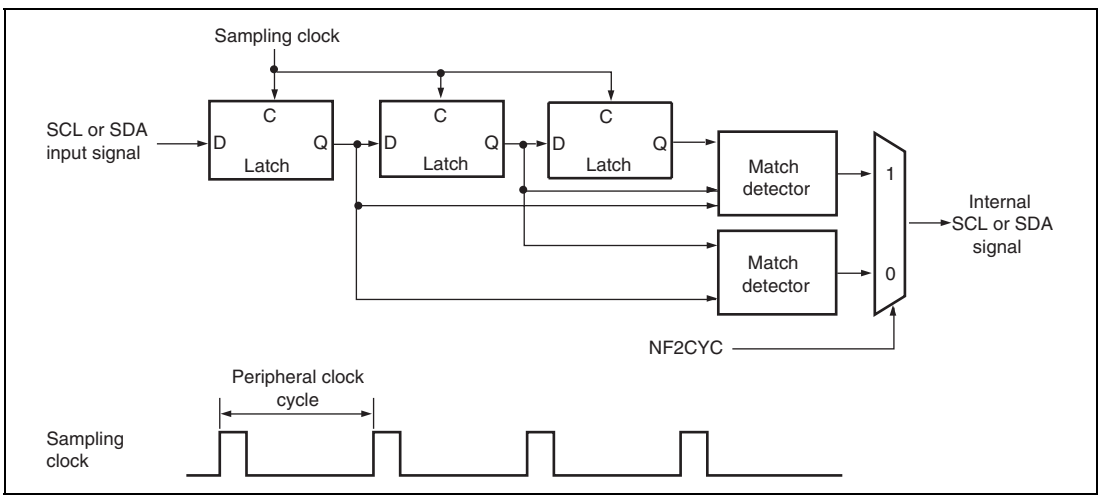



**Figure 16.17 Block Diagram of Noise Filter**

### 16.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface 2 are shown in figures 16.18 to 16.21.

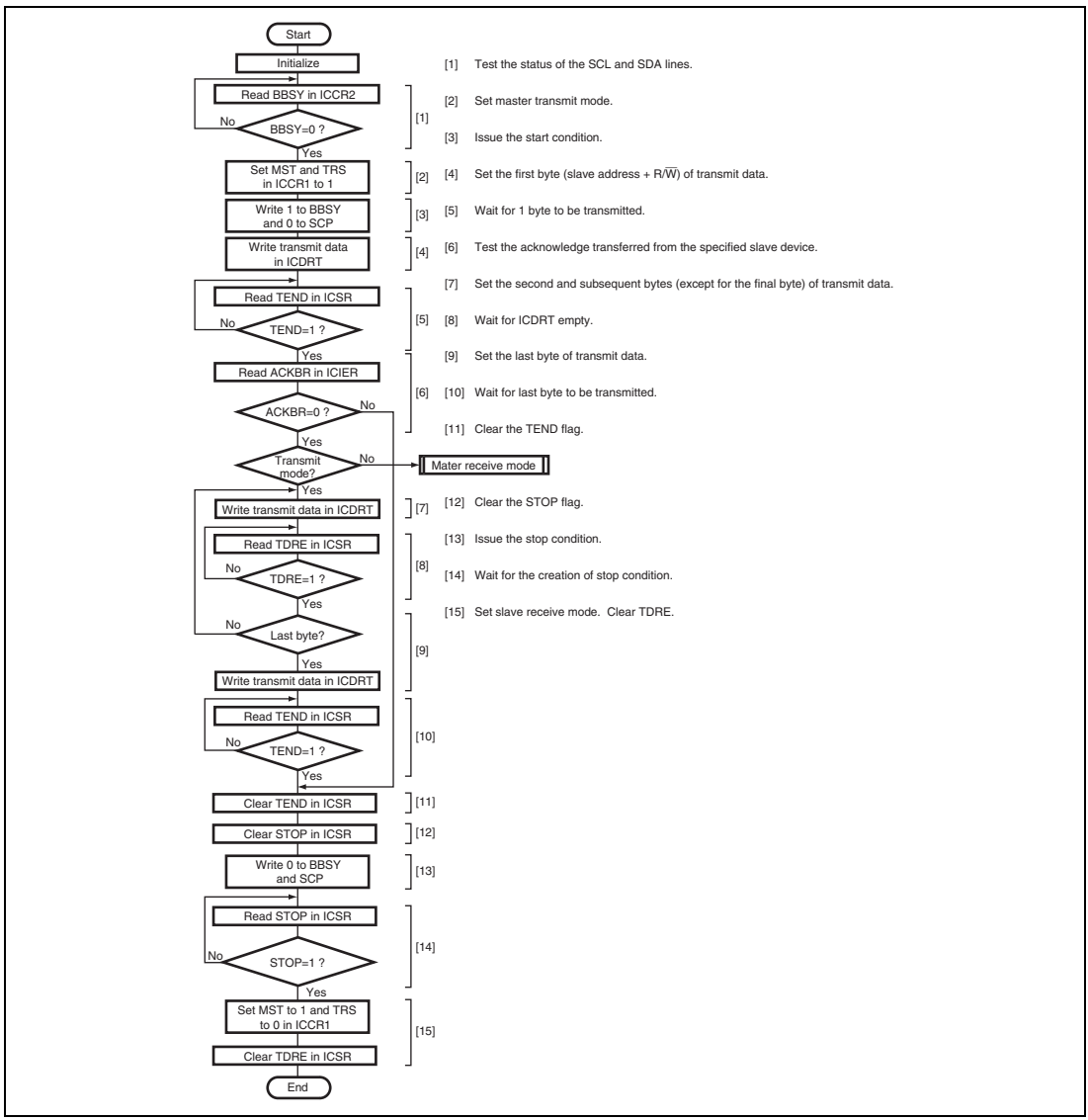
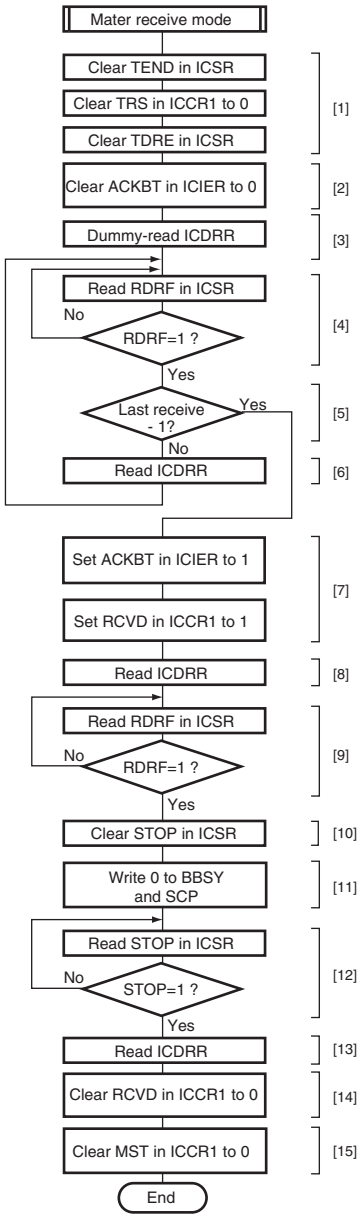



**Figure 16.18 Sample Flowchart for Master Transmit Mode**




[1] Clear TEND, select master receive mode, and then clear TDRE. *

[2] Set acknowledge to the transmit device. *

[3] Dummy-read ICDDR. *

[4] Wait for 1 byte to be received

[5] Check whether it is the (last receive - 1).

[6] Read the receive data.

[7] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).

[8] Read the (final byte - 1) of received data.

[9] Wait for the last byte to be receive.

[10] Clear the STOP flag.

[11] Issue the stop condition.

[12] Wait for the creation of stop condition.

[13] Read the last byte of receive data.

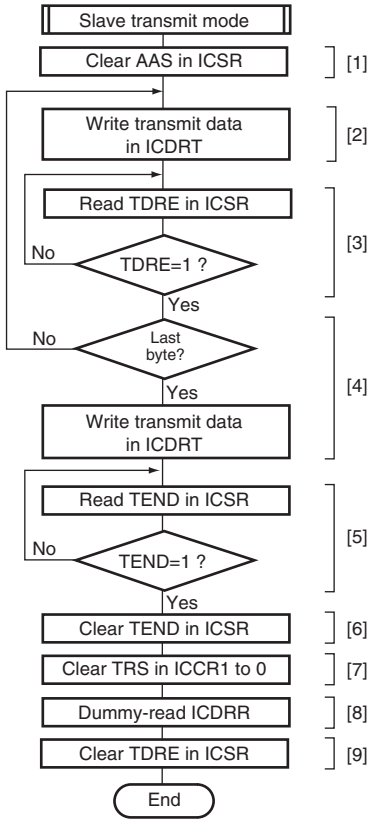
[14] Clear RCVD.

[15] Set slave receive mode.

Notes: * Make sure that no interrupt will be generated during steps [1] to [3].

When the size of receive data is only one byte in reception, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [8] is dummy-read in ICDRR.
However, when the size of receive data is two bytes and more, steps [2] to [6] are not skipped after step [1].

**Figure 16.19 Sample Flowchart for Master Receive Mode**




[1] Clear the AAS flag.

[2] Set transmit data for ICDRT (except for the last byte).

[3] Wait for ICDRT empty.

[4] Set the last byte of transmit data.

[5] Wait for the last byte to be transmitted.

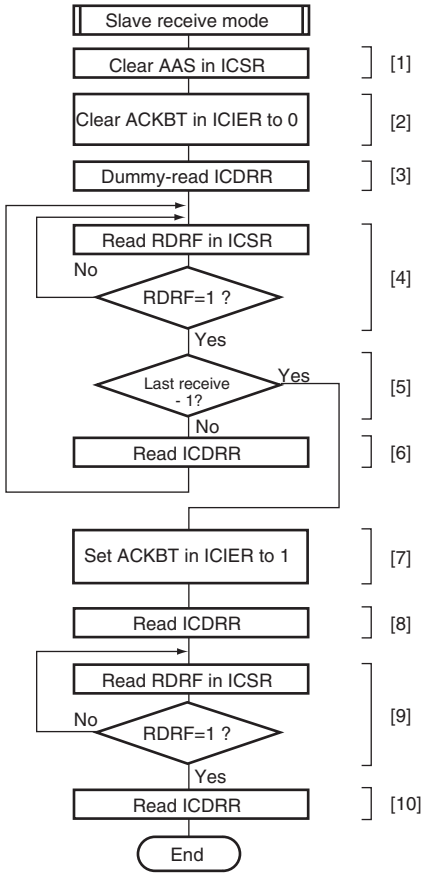
[6] Clear the TEND flag .

[7] Set slave receive mode.

[8] Dummy-read ICDRR to release the SCL line.

[9] Clear the TDRE flag.

**Figure 16.20 Sample Flowchart for Slave Transmit Mode**




[1] Clear the AAS flag.

[2] Set acknowledge to the transmit device.

[3] Dummy-read ICDRR.

[4] Wait for 1 byte to be received.

[5] Check whether it is the (last receive - 1).

[6] Read the receive data.

[7] Set acknowledge of the last byte.

[8] Read the (last byte - 1) of receive data.

[9] Wait the last byte to be received.

[10] Read for the last byte of receive data.

Note: When the size of receive data is only one byte in reception, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [8] is dummy-read in ICDRR.
However, when the size of receive data is two bytes and more, steps [2] to [6] are not skipped after step [1].

**Figure 16.21 Sample Flowchart for Slave Receive Mode**

## 16.5 I²C2 Interrupt Sources

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK receive, STOP recognition, and arbitration lost/overrun error. Table 16.4 shows the contents of each interrupt request.

**Table 16.4 Interrupt Requests**

| Interrupt Request | Abbreviation | Interrupt Condition | I²C Mode | Clock Synchronous Mode | DTC Activation |
|---|---|---|---|---|---|
| Transmit data Empty | IITXI | (TDRE=1) • (TIE=1) | √ | √ | √ |
| Transmit end | IITEI | (TEND=1) • (TEIE=1) | √ | √ | × |
| Receive data full | IIRXI | (RDRF=1) • (RIE=1) | √ | √ | √ |
| STOP recognition | IISTPI | (STOP=1) • (STIE=1) | √ | × | × |
| NACK receive | IINAKI | {(NACKF=1)+(AL=1)} • (NAKIE=1) | √ | × | × |
| Arbitration lost/ overrun error | | | √ | √ | × |

When the interrupt condition described in table 16.4 is 1, the CPU executes an interrupt exception handling. 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. The TDRE, TEND, and RDRF bits are automatically cleared while the specified number of transfers by the DTC is in progress; however, the TDRE, TEND, and RDRF bits are not cleared automatically when the transfer is complete.

## 16.6 Operation Using the DTC

In the I²C bus format, since the slave device or the direction of transfer is selected by the slave address or the R/W bit, and the acknowledge bit may indicate the end of reception or reception of the final frame, the continuous transfer of data by the DTC must be performed combined with the CPU processing by the interrupt.

Table 16.5 shows some example of processing using the DTC. These examples assume that the number of transfer data bytes is known in slave mode.

**Table 16.5 Example of Processing Using DTC**

| Item | Master Transmit Mode | Master Receive Mode | Slave Transmit Mode | Slave Receive Mode |
|---|---|---|---|---|
| Slave address + R/$\overline{W}$ bit transmission/ reception | Transmission by DTC (ICDR write) | Transmission by CPU (ICDR write) | Reception by CPU (ICDR read) | Reception by CPU (ICDR read) |
| Dummy data read | — | Processing by CPU (ICDR read) | — | — |
| Actual data transmission/ reception | Transmission by DTC (ICDR write) | Reception by DTC (ICDR read) | Transmission by DTC (ICDR write) | Reception by DTC (ICDR read) |
| Last frame processing | Not necessary | Reception by CPU (ICDR read) | Not necessary | Reception by CPU (ICDR read) |
| Setting of number of DTC transfer data frames | Transmission: Actual data count + 1 (+ 1 equivalent to slave address + R/$\overline{W}$ bits) | Reception: Actual data count | Transmission: Actual data count | Reception: Actual data count |

## 16.7 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 16.22 shows the timing of the bit synchronous circuit and table 16.6 shows the time when SCL output changes from low to Hi-Z then SCL is monitored.

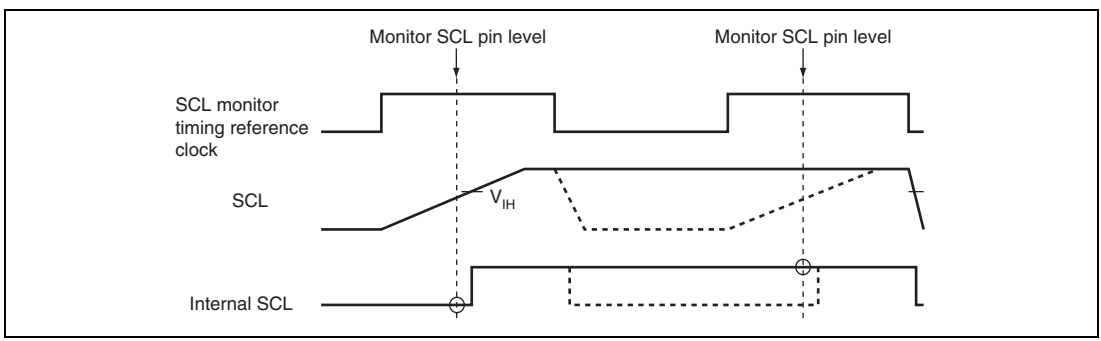



**Figure 16.22 The Timing of the Bit Synchronous Circuit**

**Table 16.6 Time for Monitoring SCL**

| CKS3 | CKS2 | NF2CYC | Time for Monitoring SCL*1 |
|---|---|---|---|
| 0 | 0 | 0 | 6.5 $t_{pcyc}$ *2 |
| | | 1 | 5.5 $t_{pcyc}$ *2 |
| | 1 | 0 | 18.5 $t_{pcyc}$ *2 |
| | | 1 | 17.5 $t_{pcyc}$ *2 |
| 1 | 0 | 0 | 16.5 $t_{pcyc}$ *2 |
| | | 1 | 15.5 $t_{pcyc}$ *2 |
| | 1 | 0 | 40.5 $t_{pcyc}$ *2 |
| | | 1 | 39.5 $t_{pcyc}$ *2 |

Notes: 1. SCL pin level is monitored after "time for monitoring SCL" has elapsed from the rising edge of the reference clock for monitoring SCL.

2. $t_{pcyc}$ indicates the period of the peripheral clock.

## 16.8 Usage Note

### 16.8.1 Module Standby Mode Setting

The I²C2 operation can be disabled or enabled using the standby control register. The initial setting is for I²C2 operation to be halted. Access to registers is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes.

### 16.8.2 Issuance of Stop Condition and Repeated Start Condition

A stop condition or repeated start condition should be issued after the fall of the ninth clock pulse is recognized. The fall of the ninth clock pulse can be recognized by checking the SCLO bit in the I²C bus control register 2 (ICCR2). When a stop condition or repeated start condition is issued at a specific timing under the conditions 1 or 2 shown below, the condition may not be output successfully. Issuance under other than these conditions will succeed with no problem.

1. When the SCL signal did not rise within the time specified in section 16.7, Bit Synchronous Circuit, due to the load of the SCL bus (load capacitance or pull-up resistor).
2. When the bit synchronous circuit is activated because the low-level periods of the eighth and ninth clock pulses are extended by the slave device.

### 16.8.3 Issuance of a Start Condition and Stop Condition in Sequence

Do not issue a start condition and stop condition in sequence. If a start condition and stop condition are to be issued in sequence, be sure to transmit a slave address before issuing the stop condition.

### 16.8.4 Settings for Multi-Master Operation

1. Transfer rate setting

   In multi-master operation, specify a transfer rate of at least 1/1.8 of the fastest transfer rate among the other masters. For example, when the fastest of the other masters is at 400 kbps, the IIC transfer rate of this LSI must be specified as 223 kbps (= 400/1.8) or a higher rate.
2. MST and TRS bits in ICCR1

   In multi-master operation, use the MOV instruction to set the MST and TRS bits in ICCR1.
3. Loss of arbitration

   When arbitration is lost, check whether the MST and TRS bits in ICCR1 are 0. If the MST and TRS bits in ICCR1 have been set to a value other than 0, clear the bits to 0.

### 16.8.5 Reading ICDRR in Master Receive Mode

In master receive mode, read ICDRR before the rising edge of the 8th clock of SCL. If ICDRR cannot be read before the rising edge of the 8th clock so that the next round of reception proceeds with the RDRF bit in ICSR set to 1, the 8the clock is fixed low and the 9th clock is output.

If ICDRR cannot be read before the rising edge of the 8th clock of SCL, set the RCVD bit in ICRR1 to 1 so that transfer proceeds in byte units.

### 16.8.6 Supported Emulator

The E200F emulator does not support I²C2 operation. Use the E10A emulator when debugging the I²C2 operation.

# Section 17 A/D Converter (ADC)

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

## 17.1 Features

- 12-bit resolution
- Input channels

  12 channels in SH7131/ SH7136 (two independent A/D conversion modules)

  16 channels in SH7132/SH7137 (two independent A/D conversion modules)
- Fast A/D conversion

  When operating at Pϕ = 40 MHz, conversion time is 1.25 μs per channel (A/D clock = 40 MHz and conversion done in 50 states)
- Two operating modes

  — Single-cycle scan mode: Continuous A/D conversion on one to eight channels

  — Continuous scan mode: Repetitive A/D conversion on one to eight channels
- 12-bit A/D data registers

  The SH7131 and SH7136 have four registers for A/D_0 and eight registers for A/D_1, which makes a total of twelve 16-bit A/D data registers (ADDR). The SH7132 and SH7137 have eight registers for both A/D_0 and A/D_1, which makes a total of sixteen 16-bit A/D data registers (ADDR). A/D conversion results are stored in A/D data registers (ADDR) that correspond to the input channels.
- Sample-and-hold function

  A sample-and-hold circuit is built into the A/D converter of this LSI, simplifying the configuration of the external analog input circuitry. Multiple channels can be sampled simultaneously because sample-and-hold circuits can be dedicated for channels 0 to 2 and 8 to 10.

  — Group A (GrA): Analog input pins selected from channels 0, 1, and 2 can be simultaneously sampled.

  — Group B (GrB): Analog input pins selected from channels 8, 9, and 10 can be simultaneously sampled.
- Three methods for starting conversion

  Software: Setting of the ADST bit in ADCR

  Timer: TRGAN, TRG0N, TRG4AN, and TRG4BN from the MTU2
  TRGAN, TRG4AN, and TRG4BN from the MTU2S

  External trigger: $\overline{\text{ADTRG}}$ (LSI pin)

- Selectable analog input channel

  A/D conversion of a selected channel is accomplished by setting the A/D analog input channel select registers (ADANSR).
- A/D conversion end interrupt and DTC transfer function is supported

  On completion of A/D conversion, A/D conversion end interrupts (ADI_3 and ADI_4) can be generated and the DTC can be activated by ADI_3 and ADI_4.

Figure 17.1 shows a block diagram of the A/D converter.

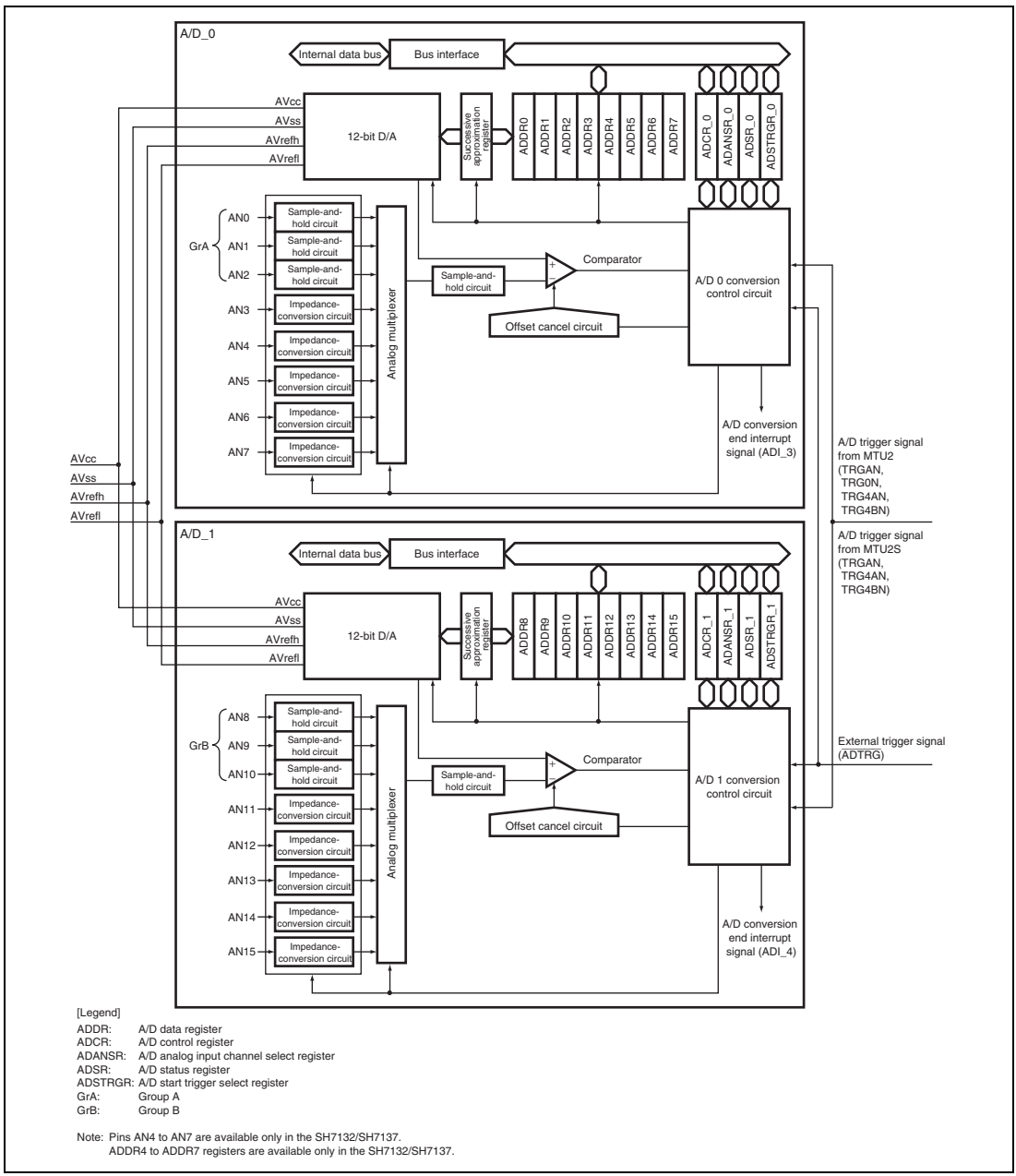



**Figure 17.1 Block Diagram of A/D Converter**

## 17.2 Input/Output Pins

Table 17.1 shows the configuration of the pins used by the A/D converter. For the pin usage, refer to the usage notes in section 17.7, Usage Notes.

**Table 17.1 Pin Configuration**

| Module Type | Pin Name | I/O | Function | Product Name | |
|---|---|---|---|---|---|
| | | | | SH7132/ SH7137 | SH7131/ SH7136 |
| Common | $AV_{CC}$ | Input | Analog block power supply pin | √ | √ |
| | $AV_{SS}$ | Input | Analog block ground pin | √ | √ |
| | $AV_{refh}$ | Input | Analog block reference power supply pin (High-side) ($AV_{refl} < AV_{refh}$) | √ | √ |
| | $AV_{refl}$ | Input | Analog block reference power supply pin (Low-side) ($AV_{refl} < AV_{refh}$) | √ | √ |
| | $\overline{ADTRG}$ | Input | A/D external trigger input pin | √ | √ |
| A/D module 0 (A/D_0) | AN0 | Input | Analog input pin 0 (Group A) | √ | √ |
| | AN1 | Input | Analog input pin 1 (Group A) | √ | √ |
| | AN2 | Input | Analog input pin 2 (Group A) | √ | √ |
| | AN3 | Input | Analog input pin 3 | √ | √ |
| | AN4 | Input | Analog input pin 4 | √ | — |
| | AN5 | Input | Analog input pin 5 | √ | — |
| | AN6 | Input | Analog input pin 6 | √ | — |
| | AN7 | Input | Analog input pin 7 | √ | — |
| A/D module 1 (A/D_1) | AN8 | Input | Analog input pin 8 (Group B) | √ | √ |
| | AN9 | Input | Analog input pin 9 (Group B) | √ | √ |
| | AN10 | Input | Analog input pin 10 (Group B) | √ | √ |
| | AN11 | Input | Analog input pin 11 | √ | √ |
| | AN12 | Input | Analog input pin 12 | √ | √ |
| | AN13 | Input | Analog input pin 13 | √ | √ |
| | AN14 | Input | Analog input pin 14 | √ | √ |
| | AN15 | Input | Analog input pin 15 | √ | √ |

## 17.3 Register Descriptions

The A/D converter has the following registers.

**Table 17.2 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| A/D control register_0 | ADCR_0 | R/W | H'00 | H'FFFFD400 | 8 |
| A/D status register_0 | ADSR_0 | R/W | H'00 | H'FFFFD402 | 8 |
| A/D start trigger select register_0 | ADSTRGR_0 | R/W | H'00 | H'FFFFD41C | 8 |
| A/D analog input channel select register_0 | ADANSR_0 | R/W | H'00 | H'FFFFD420 | 8 |
| A/D data register 0 | ADDR0 | R | H'0000 | H'FFFFD440 | 16 |
| A/D data register 1 | ADDR1 | R | H'0000 | H'FFFFD442 | 16 |
| A/D data register 2 | ADDR2 | R | H'0000 | H'FFFFD444 | 16 |
| A/D data register 3 | ADDR3 | R | H'0000 | H'FFFFD446 | 16 |
| A/D data register 4 | ADDR4 | R | H'0000 | H'FFFFD448 | 16 |
| A/D data register 5 | ADDR5 | R | H'0000 | H'FFFFD44A | 16 |
| A/D data register 6 | ADDR6 | R | H'0000 | H'FFFFD44C | 16 |
| A/D data register 7 | ADDR7 | R | H'0000 | H'FFFFD44E | 16 |
| A/D control register_1 | ADCR_1 | R/W | H'00 | H'FFFFD600 | 8 |
| A/D status register_1 | ADSR_1 | R/W | H'00 | H'FFFFD602 | 8 |
| A/D start trigger select register_1 | ADSTRGR_1 | R/W | H'00 | H'FFFFD61C | 8 |
| A/D analog input channel select register_1 | ADANSR_1 | R/W | H'00 | H'FFFFD620 | 8 |
| A/D data register 8 | ADDR8 | R | H'0000 | H'FFFFD640 | 16 |
| A/D data register 9 | ADDR9 | R | H'0000 | H'FFFFD642 | 16 |
| A/D data register 10 | ADDR10 | R | H'0000 | H'FFFFD644 | 16 |
| A/D data register 11 | ADDR11 | R | H'0000 | H'FFFFD646 | 16 |
| A/D data register 12 | ADDR12 | R | H'0000 | H'FFFFD648 | 16 |
| A/D data register 13 | ADDR13 | R | H'0000 | H'FFFFD64A | 16 |
| A/D data register 14 | ADDR14 | R | H'0000 | H'FFFFD64C | 16 |
| A/D data register 15 | ADDR15 | R | H'0000 | H'FFFFD64E | 16 |

### 17.3.1 A/D Control Registers_0 and _1 (ADCR_0 and ADCR_1)

ADCRs are 8-bit readable/writable registers that select conversion mode for the A/D_0 and A/D_1.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADST | ADCS | ACE | ADIE | - | - | TRGE | EXTRG |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | ADST | 0 | R/W | A/D Start<br>When this bit is cleared to 0, A/D conversion is stopped and the A/D converter enters the idle state. When this bit is set to 1, A/D conversion is started. In single-cycle scan mode, this bit is automatically cleared to 0 when A/D conversion ends on the selected single channel. In continuous scan mode, A/D conversion is continuously performed for the selected channels in sequence until this bit is cleared by software, a reset, software standby mode, or module standby mode. |
| 6 | ADCS | 0 | R/W | A/D Continuous Scan<br>Selects either a single-cycle or a continuous scan in scan mode. This bit is valid only when scan mode is selected.<br>0: Single-cycle scan<br>1: Continuous scan<br>When changing the operating mode, first clear the ADST bit to 0. |
| 5 | ACE | 0 | R/W | Automatic Clear Enable<br>Enables or disables the automatic clearing of ADDR after ADDR is read by the CPU or DTC. When this bit is set to 1, ADDR is automatically cleared to H'0000 after the CPU or DTC reads ADDR. This function allows the detection of any renewal failures of ADDR.<br>0: Automatic clearing of ADDR after being read is disabled.<br>1: Automatic clearing of ADDR after being read is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | ADIE | 0 | R/W | A/D Interrupt Enable<br>Enables or disables the generation of A/D conversion end interrupts (ADI_3 and ADI_4) to the CPU. Operating modes must be changed when the ADST bit is 0 to prevent incorrect operations.<br>When A/D conversion ends and the ADF bit in ADSR is set to 1 and this bit is set to 1, ADI_3 or ADI_4 is sent to the CPU. By clearing the ADF bit or the ADIE bit to 0, ADI_3 and ADI_4 can be cleared.<br>0: Generation of A/D conversion end interrupt is disabled<br>1: Generation of A/D conversion end interrupt is enabled |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | TRGE | 0 | R/W | Trigger Enable<br>Enables or disables A/D conversion start by the external trigger input ($\overline{\text{ADTRG}}$) or A/D conversion start triggers from the MTU2 and MTU2S (TRGAN, TRG0N, TRG4AN, and TRG4BN from the MTU2 and TRGAN, TRG4AN, and TRG4BN from the MTU2S). For selection of the external trigger and A/D conversion start trigger from the MTU2 or MTU2S, see the description of the EXTRG bit.<br>0: A/D conversion start by the external trigger or an A/D conversion start trigger from the MTU or MTU2S is disabled<br>1: A/D conversion start by the external trigger or an A/D conversion start trigger from the MTU2 or MTU2S is enabled |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | EXTRG | 0 | R/W | Trigger Select<br>Selects the external trigger ($\overline{\text{ADTRG}}$) or an A/D conversion start trigger from the MTU2 or MTU2S as an A/D conversion start trigger.<br>When the external trigger is selected (EXTRG = 1), upon input of a low-level pulse to the $\overline{\text{ADTRG}}$ pin after the TRGE bit is set to 1, the A/D converter detects the falling edge of the pulse, and sets the ADST bit in ADCR to 1. The operation which is performed when 1 is written to the ADST bit by software is subsequently performed. A/D conversion start by the external trigger input is enabled only when the ADST bit is cleared to 0.<br>When the external trigger is used as an A/D conversion start trigger, the low-level pulse input to the $\overline{\text{ADTRG}}$ pin must be at least 1.5 Pϕ clock cycles in width.<br>0: A/D converter is started by the A/D conversion start trigger from the MTU2 or MTU2S<br>1: A/D converter is started by the external pin ($\overline{\text{ADTRG}}$) |

### 17.3.2 A/D Status Registers_0 and _1 (ADSR_0 and ADSR_1)

ADSRs are 8-bit readable/writable registers that indicate the status of the A/D converter.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | ADF |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R/(W)* |

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way. Do not overwrite this bit with 0 when the value of this bit is 0.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | ADF | 0 | R/(W)* | A/D End Flag<br>A status flag that indicates the completion of A/D conversion.<br>[Setting condition]<br>• When A/D conversion on all specified channels is completed in scan mode<br>[Clearing conditions]<br>• When 0 is written after reading ADF = 1<br>• When the DTC is activated by an ADI interrupt and ADDR is read |

### 17.3.3 A/D Start Trigger Select Registers_0 and _1 (ADSTRGR_0 and ADSTRGR_1)

ADSTRGRs select an A/D conversion start trigger from the MTU2 or MTU2S. The A/D conversion start trigger is used as an A/D conversion start source when the TRGE bit in ADCR is set to 1 and the EXTRG bit in ADCR is set to 0.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | STR6 | STR5 | STR4 | STR3 | STR2 | STR1 | STR0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | STR6 | 0 | R/W | Start Trigger 6<br>Enables or disables the A/D conversion start request input from the MTU2S.<br>0: Disables the A/D conversion start by TRGAN trigger (MTU2S).<br>1: Enables the A/D conversion start by TRGAN trigger (MTU2S). |
| 5 | STR5 | 0 | R/W | Start Trigger 5<br>Enables or disables the A/D conversion start request input from the MTU2S.<br>0: Disables the A/D conversion start by TRG4AN trigger (MTU2S).<br>1: Enables the A/D conversion start by TRG4AN trigger (MTU2S). |
| 4 | STR4 | 0 | R/W | Start Trigger 4<br>Enables or disables the A/D conversion start request input from the MTU2S.<br>0: Disables the A/D conversion start by TRG4BN trigger (MTU2S).<br>1: Enables the A/D conversion start by TRG4BN trigger (MTU2S). |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | STR3 | 0 | R/W | Start Trigger 3<br>Enables or disables the A/D conversion start request input from the MTU2.<br>0: Disables the A/D conversion start by TRG0N trigger (MTU2).<br>1: Enables the A/D conversion start by TRG0N trigger (MTU2). |
| 2 | STR2 | 0 | R/W | Start Trigger 2<br>Enables or disables the A/D conversion start request input from the MTU2.<br>0: Disables the A/D conversion start by TRGAN trigger (MTU2).<br>1: Enables the A/D conversion start by TRGAN trigger (MTU2). |
| 1 | STR1 | 0 | R/W | Start Trigger 1<br>Enables or disables the A/D conversion start request input from the MTU2.<br>0: Disables the A/D conversion start by TRG4AN trigger (MTU2).<br>1: Enables the A/D conversion start by TRG4AN trigger (MTU2). |
| 0 | STR0 | 0 | R/W | Start Trigger 0<br>Enables or disables the A/D conversion start request input from the MTU2.<br>0: Disables the A/D conversion start by TRG4BN trigger (MTU2).<br>1: Enables the A/D conversion start by TRG4BN trigger (MTU2). |

### 17.3.4 A/D Analog Input Channel Select Registers_0 and _1 (ADANSR_0 and ADANSR_1)

ADANSRs are 8-bit readable/writable registers that select an analog input channel.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | ANS7 | 0 | R/W | Setting bits in the A/D analog input channel select register to 1 selects a channel that corresponds to a specified bit. For the correspondence between analog input pins and bits, see table 17.3.<br>When changing the analog input channel, the ADST bit in ADCR must be cleared to 0 to prevent incorrect operations. |
| 6 | ANS6 | 0 | R/W | |
| 5 | ANS5 | 0 | R/W | |
| 4 | ANS4 | 0 | R/W | |
| 3 | ANS3 | 0 | R/W | |
| 2 | ANS2 | 0 | R/W | |
| 1 | ANS1 | 0 | R/W | |
| 0 | ANS0 | 0 | R/W | |

**Table 17.3 Channel Select List**

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

### 17.3.5 A/D Data Registers 0 to 15 (ADDR0 to ADDR15)

ADDRs are 16-bit read-only registers. The conversion result for each analog input channel is stored in ADDR with the corresponding number. (See table 17.4.)

The converted 12-bit data is stored in bits 11 to 0.

The initial value of ADDR is H'0000.

After ADDR is read, ADDR can be automatically cleared to H'0000 by setting the ACE bit in ADCR to 1.

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 12 | — | All 0 | R | Reserved |
| 11 to 0 | ADD[11:0] | All 0 | R | 12-bit data |

**Table 17.4 Correspondence between Analog Channels and Registers (ADDR0 to ADDR15)**

| A/D_0 Converter | | A/D_1 Converter | |
|---|---|---|---|
| **Analog Input Channels** | **A/D Data Registers** | **Analog Input Channels** | **A/D Data Registers** |
| AN0 | ADDR0 | AN8 | ADDR8 |
| AN1 | ADDR1 | AN9 | ADDR9 |
| AN2 | ADDR2 | AN10 | ADDR10 |
| AN3 | ADDR3 | AN11 | ADDR11 |
| AN4 | ADDR4 | AN12 | ADDR12 |
| AN5 | ADDR5 | AN13 | ADDR13 |
| AN6 | ADDR6 | AN14 | ADDR14 |
| AN7 | ADDR7 | AN15 | ADDR15 |

### 17.3.6 CPU Interface

Since the internal bus connected to the CPU is 16 bits wide, the upper and lower bytes of data can be read simultaneously.

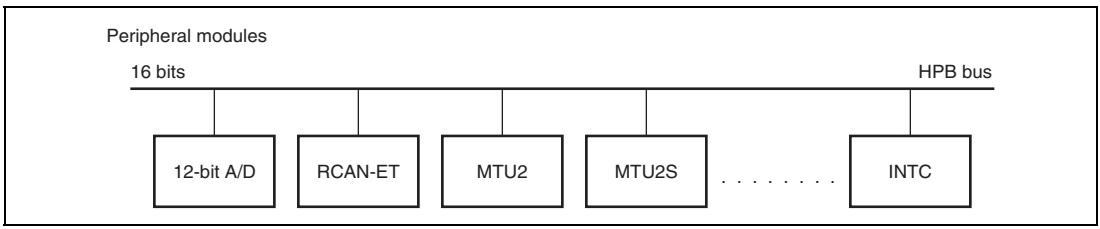



**Figure 17.2 Interface Between CPU and 12-Bit A/D Converter**

## 17.4 Operation

The A/D converter has two operating modes: single-cycle scan mode and continuous scan mode. In single-cycle scan mode, A/D conversion is performed once on one or more specified channels and then it ends. In continuous scan mode, the A/D conversion is performed sequentially on one or more specified channels until the ADST bit is cleared to 0.

The ADCS bit in the A/D control register (ADCR) is used to select the operating mode. Setting the ADCS bit to 0 selects single-cycle scan mode and setting the ADCS bit to 1 selects continuous scan mode. In both modes, A/D conversion starts on the channel with the lowest number in the analog input channels selected by the A/D analog input channel select register (ADANSR). The A/D_0 performs conversions from AN0 to AN7 and A/D_1 from AN8 to AN15.

In single-cycle scan mode, when one cycle of A/D conversion on all specified channels is completed, the ADF bit in ADSR is set to 1 and the ADST bit is automatically cleared to 0. In continuous scan mode, when conversion on all specified channels is completed, the ADF bit in ADSR is set to 1. To stop A/D conversion, write 0 to the ADST bit. When the ADF bit is set to 1, if the ADIE bit in ADCR is set to 1, an A/D conversion end interrupt (ADI) is generated. When clearing the ADF bit to 0, read the ADF bit while set to 1 and then write 0. However, when the DTC is activated by an ADI interrupt, the ADF bit is automatically cleared to 0.

### 17.4.1 Single-Cycle Scan Mode

The following example shows the operation when analog input channels 0 to 3 (AN0 to AN3) are selected and the A/D_0 conversion is performed in single-cycle scan mode using four channels. This operation also applies to the A/D_1 conversion.

1. Set the ADCS bit in the A/D control register_0 (ADCR_0) to 0.
2. Set all bits ANS0 to ANS3 in the A/D analog input channel select register_0 (ADANSR_0) to 1.
3. Set the ADST bit in the A/D control register_0 (ADCR_0) to 1 to start A/D conversion.
4. After channels 0 to 2 (GrA) are sampled simultaneously, offset canceling processing (OFC) is performed. Then, A/D conversion is performed on channel 0. Upon completion of the A/D conversion, the A/D conversion result is transferred to ADDR0. Following this, channel 1 is converted. Upon completion of the conversion, the A/D conversion result is transferred to ADDR1. In the same way, channel 2 is converted and the A/D conversion result is transferred to ADDR2.

   A/D conversion of channel 3 is then started. Upon completion of the A/D conversion, the A/D conversion result is transferred to ADDR3.

5. When A/D conversion ends on all specified channels (AN0 to AN3), the ADF bit is set to 1, the ADST bit is automatically cleared to 0, and the A/D conversion ends. At this time, if the ADIE bit is set to 1, an ADI_3 interrupt is generated after the A/D conversion.

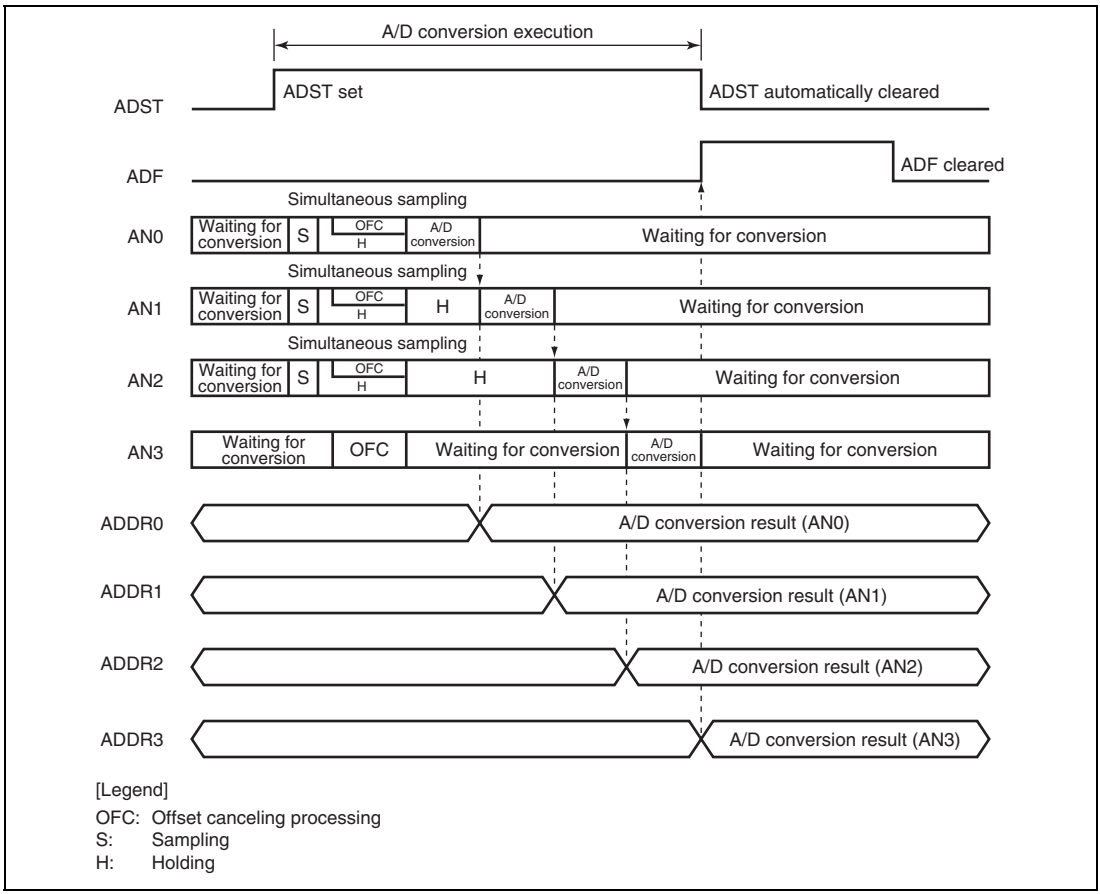



**Figure 17.3 Example of A/D_0 Converter Operation (Single-Cycle Scan Mode)**

### 17.4.2 Continuous Scan Mode

The following example shows the operation when analog input channels 0, 2, and 3 (AN0, AN2, AN3) are selected and the A/D_0 conversion is performed in continuous scan mode using the three channels. This operation also applies to the A/D_1 conversion.

1. Set the ADCS bit in the A/D control register_0 (ADCR_0) to 0.
2. Set all bits ANS0, ANS2, and ANS3 in the A/D analog input channel select register_0 (ADANSR_0) to 1.
3. Set the ADST bit in the A/D control register_0 (ADCR_0) to 1 to start A/D conversion.
4. Channels 0 and 2 (GrA) are sampled simultaneously. As the ANS1 bit in ADANSR_0 is set to 0, channel 1 is not sampled. After this, offset canceling processing (OFC) is performed. Then the A/D conversion on channel 0 is started. Upon completion of the A/D conversion, the A/D conversion result is transferred to ADDR0. In the same way, channel 2 is converted and the A/D conversion result is transferred to ADDR2. The A/D conversion is not performed on channel 1.
5. The A/D conversion of channel 3 is started. Upon completion of the A/D conversion, the A/D conversion result is transferred to ADDR3.
6. When the A/D conversion ends on all the specified channels (AN0 to AN3), the ADF bit is set to 1. At this time, if the ADIE bit is set to 1, an ADI_3 interrupt is generated after the A/D conversion.
7. Steps 4 to 6 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, the A/D conversion stops. After this, if the ADST bit is set to 1, the A/D conversion starts again and repeats steps 4 to 6.

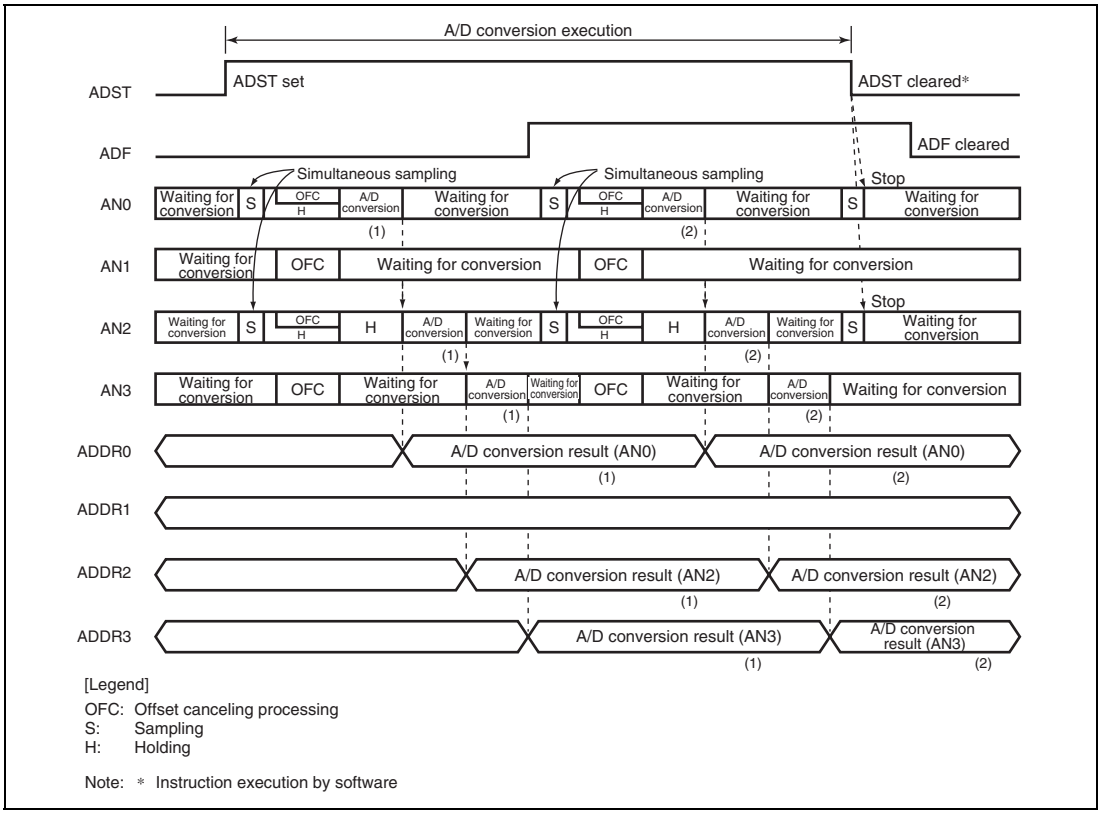



**Figure 17.4 Example of A/D_0 Converter Operation (Continuous Scan Mode)**

### 17.4.3 Input Sampling and A/D Conversion Time

The A/D_0 has a built-in sample-and-hold circuit common to all the channels. Each of channels 0 to 2 of the A/D_0 has a dedicated built-in sample-and-hold circuit. Channels 0 to 2 can be simultaneously sampled as one group. This group is referred to as Group A (GrA) (in table 17.5). Even when only one channel is selected in the group by ADANSR, the sample-and-hold operation is performed with the dedicated sample-and-hold circuit. When only the channels without a dedicated sample-and-hold circuit are specified by ADANSR, the time that elapses is the same as when a dedicated sample-and-hold circuit is used.

The above descriptions is the same with the A/D_1.

When an event that sets the ADST bit writing to this bit by the CPU, A/D converter activation request from the MTU2, the MTU2S, and an external trigger signal occurs, the analog input is sampled by the dedicated sample-and-hold circuit for each channel after the A/D conversion start delay time ($t_D$) has passed and the offset canceling processing (OFC) is performed. After this, the sampling of the analog input using the sample-and-hold circuit common to all the channels is performed and then the A/D conversion is started. Figure 17.5 shows the A/D conversion timing in this case. This A/D conversion time ($t_{CONV}$) includes the $t_D$, the offset canceling processing time ($t_{OFC}$), the analog input sampling time with a dedicated sample-and-hold circuit for each channel ($t_{SPLSH}$), and the analog input sampling time with the sample-and-hold circuit common to all the channels ($t_{SPL}$). The $t_{SPLSH}$ does not depend on the number of channels simultaneously sampled.

In continuous scan mode, the A/D conversion time ($t_{CONV}$) given in table 17.6 applies to the conversion time of the first cycle. The conversion time of the second and subsequent cycles is expressed as ($t_{CONV} - t_D + 6$).

**Table 17.5 Correspondence between Analog Input Channels and Groups being Allowed Simultaneous Sampling**

| A/D_0 Converter | | A/D_1 Converter | |
|---|---|---|---|
| **Analog Input Channels** | **Group** | **Analog Input Channels** | **Group** |
| AN0 | GrA | AN8 | GrB |
| AN1 | | AN9 | |
| AN2 | | AN10 | |
| AN3 | — | AN11 | — |
| AN4 | — | AN12 | — |
| AN5 | — | AN13 | — |
| AN6 | — | AN14 | — |
| AN7 | — | AN15 | — |

**Table 17.6 A/D Conversion Time**

| | | Number of Required States | | |
|---|---|---|---|---|
| **Item** | **Symbol** | **Min.** | **Typ.** | **Max.** |
| A/D conversion start delay time | $t_D$ | 11*[1] | — | 15*[2] |
| Analog input sampling time of dedicated sample-and-hold circuit for GrA and GrB | $t_{SPLSH}$ | — | 30 | — |
| Offset canceling processing time | $t_{OFC}$ | — | 50 | — |
| Analog input sampling time of sample-and-hold circuit common to all channels | $t_{SPL}$ | — | 20 | — |
| A/D conversion time | $t_{CONV}$ | 50n + 95*[3] | — | 50n + 99*[3] |

Notes: 1. A/D converter activation by the MTU2 or MTU2S trigger signal.
2. A/D converter activation by an external trigger signal.
3. n: number of A/D conversion channels (n = 1 to 8)

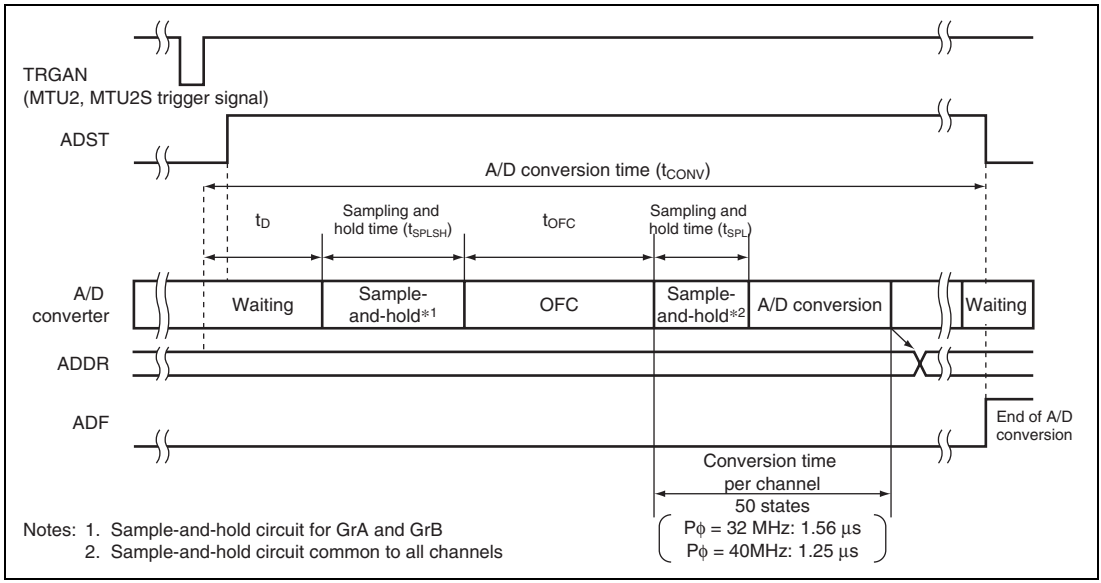



**Figure 17.5 A/D Conversion Timing (Single-Cycle Scan Mode)**

### 17.4.4 A/D Converter Activation by MTU2 and MTU2S

A/D conversion is activated by the A/D conversion start triggers (TRGAN, TRG0N, TRG4N, and TRG4BN) from the MTU2 and A/D conversion start triggers (TRGAN, TRG4AN, and TRG4BN) from the MTU2S. To enable this function, set the TRGE bit in ADCR to 1 and clear the EXTRG bit to 0. After this setting is made, if an A/D conversion start trigger from the MTU2 or MTU2S is generated, the ADST bit is set to 1. The timing between the setting of the ADST bit and the start of the A/D conversion is the same for all A/D conversion activation soures.

The A/D conversion start trigger must be input after ADCR, ADSTRGR, and ADANSR registers have been set.

### 17.4.5 External Trigger Input Timing

The A/D conversion can be externally triggered. To input an external trigger, set the pin function controller (PFC) to select $\overline{\text{ADTRG}}$ pin function and drive the $\overline{\text{ADTRG}}$ pin low when a high level is input to the $\overline{\text{ADTRG}}$ pin with the TRGE and EXTRG bits in ADCR are both set to 1. A falling edge of the $\overline{\text{ADTRG}}$ pin sets the ADST bit in ADCR to 1, starting the A/D conversion. Other operations are conducted in the same way for all A/D conversion activation soures. Figure 17.6 shows the timing.

The ADST bit is set to 1 after 5 states has elapsed from the point at which the A/D converter detects a falling edge on the $\overline{\text{ADTRG}}$ pin. A low level input to the $\overline{\text{ADTRG}}$ pin must be made after the ADCR, ADSTRGR, and ADANSR registers have been set.

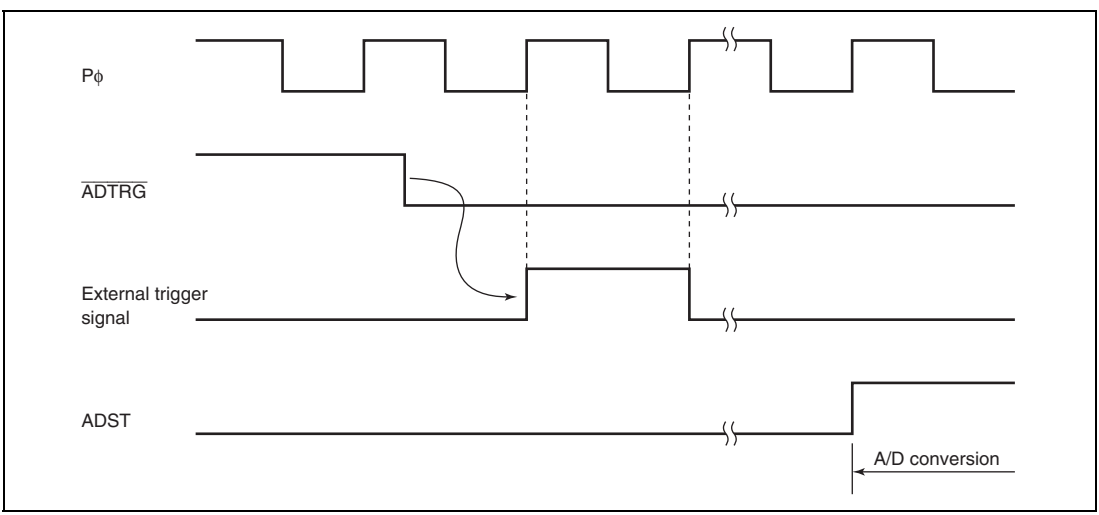



**Figure 17.6 External Trigger Input Timing**

### 17.4.6 Example of ADDR Auto-Clear Function

When the A/D data register (ADDR) is read by the CPU or DTC, ADDR can be automatically cleared to H'0000 by setting the ACE bit in ADCR to 1. This function allows the detection of an ADDR renewal failure.

Figure 17.7 shows an example of when the auto-clear function of ADDR is disabled (normal state) and enabled.

When the ACE bit is 0 (initial value) and the A/D conversion result (H'0222) is not written to ADDR for some reason, the old data (H'0111) becomes the ADDR value. In addition, when the ADDR value is read into a general register using an A/D conversion end interrupt, the old data (H'0111) is stored in the general register. To detect a renewal failure, every time the old data needs to be stored in the RAM, a general register, etc.

When the ACE bit is 1, reading ADDR = H'0111 by the CPU or DTC automatically clears ADDR to H'0000. After this, if the A/D conversion result (H'0222) cannot be transferred to ADDR for some reason, the cleared data (H'0000) remains as the ADDR value. When this ADDR value is read into a general register, H'0000 is stored in the general register. Just by checking whether the read data value is H'0000 or not allows the detection of an ADDR renewal failure.

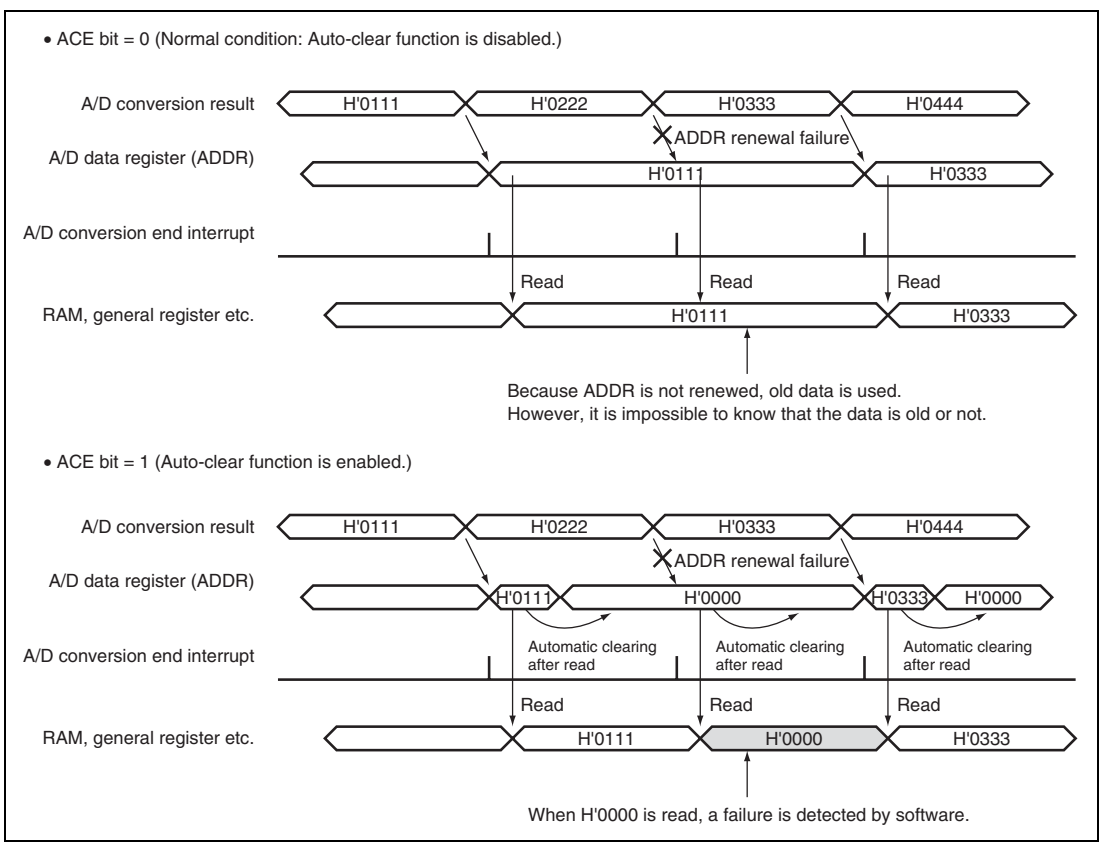



**Figure 17.7 Example of When ADDR Auto-clear Function is Disabled (Normal Condition)/Enabled**

## 17.5 Interrupt Sources and DTC Transfer Requests

The A/D converter generates A/D conversion end interrupts (ADI_3 and ADI_4). Table 17.7 lists the interrupt sources. An ADI_3 interrupt generation is enabled when the ADIE bit in ADCR_0 is set to 1. An ADI_4 interrupt generation is enabled when the ADIE bit in ADCR_1 is set to 1. On the other hand, an ADI_3 interrupt generation is disabled when the ADIE bit in ADCR_0 is cleared to 0, and an ADI_4 interrupt generation is disabled when the ADIE bit in ADCR_1 is cleared to 0. The data transfer controller (DTC) can be activated by the DTC setting when an ADI_3 or ADI_4 interrupt is generated. When the DTC is activated by an ADI_3 or an ADI_4 interrupt, the ADF bit in ADSR_0 and ADSR_1 is automatically cleared.

**Table 17.7 Interrupt Sources**

| Channel | Interrupt Source | Interrupt Enable Bit | Interrupt Flag Bit | DTC Activation | Priority |
|---|---|---|---|---|---|
| A/D_0 | A/D_3 | ADIE | ADF | Possible | High |
| A/D_1 | A/D_4 | ADIE | ADF | Possible | Low |

## 17.6 Definitions of A/D Conversion Accuracy

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital conversion output codes
- Offset error

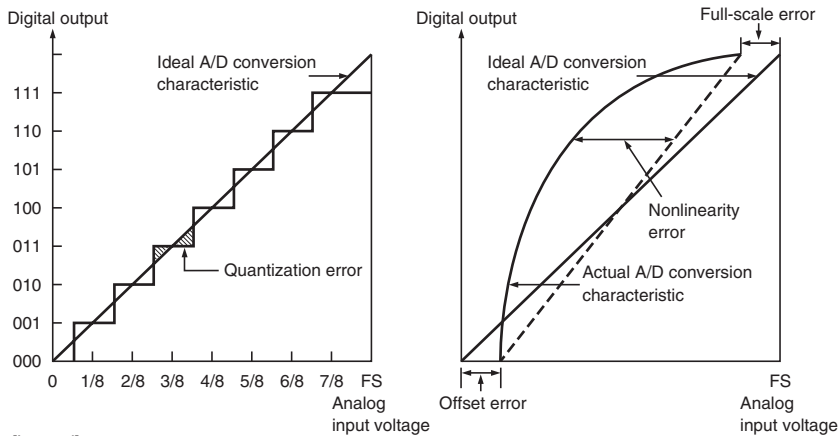
  The deviation of the actual A/D conversion characteristic from the ideal A/D conversion characteristic when the digital output value changes from the minimum voltage value (zero voltage) B'000000000000 to B'000000000001. Does not include a quantization error (see figure 17.8).
- Full-scale error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conversion characteristic when the digital output value changes from B'111111111110 to the maximum voltage value (full-scale voltage) B'111111111111. Does not include a quantization error (see figure 17.8).
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17.8).
- Nonlinearity error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conversion characteristic between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error (see figure 17.8).
- Absolute accuracy

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




[Legend]
FS: Full-scale

**Figure 17.8 Definitions of A/D Conversion Accuracy**

## 17.7 Usage Notes

### 17.7.1 Analog Input Voltage Range

The voltage applied to analog input pin (ANn) during A/D conversion should be in the range $AV_{refl} \leq ANn$ (n = 0 to 15) $\leq AV_{refh}$.

### 17.7.2 Relationship between AVcc, AVss and Vcc, Vss

When using the A/D converter, set AVcc = 5.0 V ±0.5 V and AVss = Vss. When the A/D converter is not used, set AVss = Vss, and do not leave the AVcc pin open.

### 17.7.3 Range of $AV_{refh}$ and $AV_{refl}$ Pin Settings

When using the A/D converter, set $AV_{refh}$ = 4.5 to AVcc. When the A/D converter is not used, set $AV_{refh} \leq$ AVcc. If these conditions are not met, the reliability of the LSI may be adversely affected. For $AV_{refl}$, set $AV_{refl}$ = AVss = Vss.

### 17.7.4 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and the layout in which the digital circuit signal lines and analog circuit signal lines cross or are in close proximity to each other should be avoided as much as possible. Failure to do so may result in the incorrect operation of the analog circuitry due to inductance, adversely affecting the A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN15), analog reference power supply ($AV_{refh}$ and $AV_{refl}$), the analog power supply (AVcc), and the analog ground (AVss). Also, AVss should be connected at one point to a stable digital ground (Vss) on the board.

### 17.7.5 Notes on Noise Countermeasures

To prevent damage due to an abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN15) and analog reference power supply ($AV_{refh}$, $AV_{refl}$), a protection circuit should be connected between the AVcc and AVss, as shown in figure 17.9. Also, the bypass capacitors connected to $AV_{refh}$ and $AV_{refl}$ and the filter capacitor connected to ANn should be connected to the AVss. If a filter capacitor is connected as shown in figure 17.9, the input currents at the analog input pin (ANn) are averaged, and an error may occur. Careful consideration is therefore required when deciding the circuit constants.

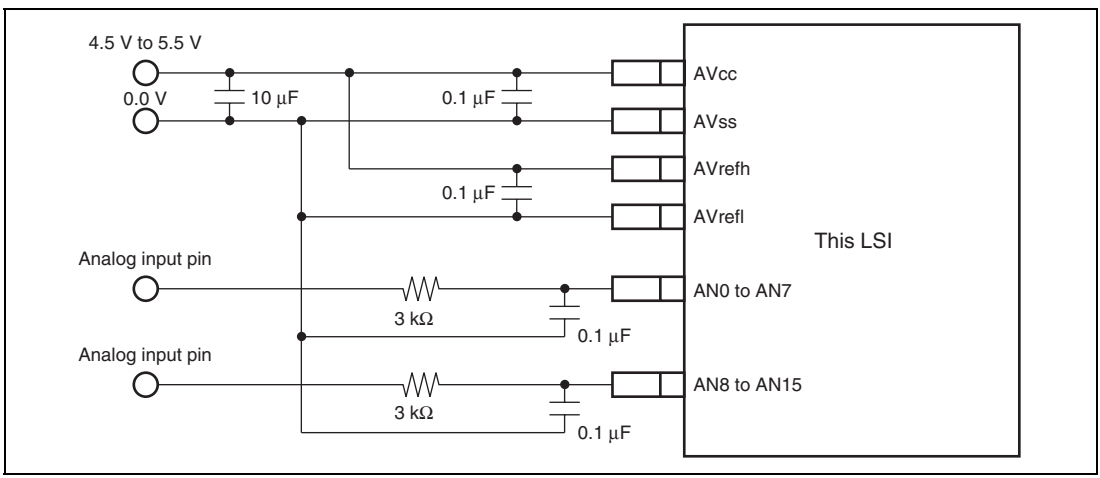



**Figure 17.9 Example of Analog Input Pin Protection Circuit**

### 17.7.6 Notes on Register Setting

- Set the ADST bit in the A/D control register (ADCR) after the A/D start trigger select register (ADSTRGR) and the A/D analog input channel select register (ADANSR) have been set. Do not modify the settings of the ADCS, ACE, ADIE, TRGE, and EXTRG bits while the ADST bit in the ADCR register is set to 1.
- Do not start the A/D conversion when the ANS bits (ANS[7:0]) in the A/D analog input channel select register (ADANSR) are all 0.

# Section 18 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer (CMT) consisting of a 2-channel 16-bit timer. The CMT has a16-bit counter, and can generate interrupts at set intervals.

## 18.1 Features

- Selection of four counter input clocks

  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) can be selected independently for each channel.
- Interrupt request on compare match
- Module standby mode can be set.

Figure 18.1 shows a block diagram of CMT.

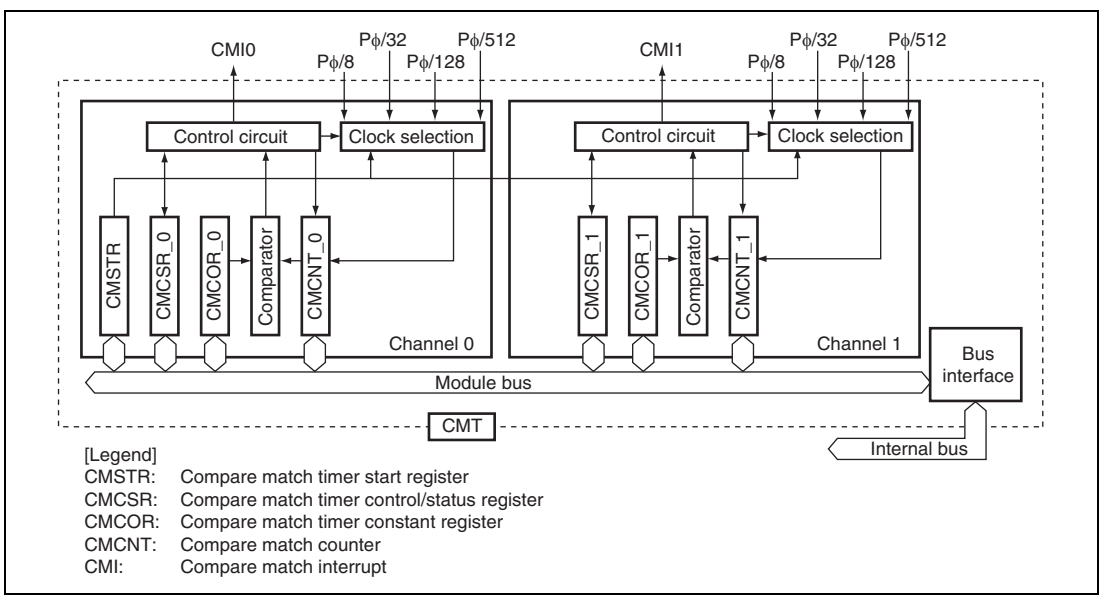



**Figure 18.1 Block Diagram of CMT**

## 18.2 Register Descriptions

The CMT has the following registers. For details on register addresses and register states during each processing, refer to section 25, List of Registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name.

**Table 18.1 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Compare match timer start register | CMSTR | R/W | H'0000 | H'FFFFCE00 | 8, 16, 32 |
| Compare match timer control/status register_1 | CMCSR_0 | R/W | H'0000 | H'FFFFCE02 | 8, 16 |
| Compare match counter_0 | CMCNT_0 | R/W | H'0000 | H'FFFFCE04 | 8, 16, 32 |
| Compare match constant register_0 | CMCOR_0 | R/W | H'FFFF | H'FFFFCE06 | 8, 16 |
| Compare match timer control/status register_0 | CMCSR_1 | R/W | H'0000 | H'FFFFCE08 | 8, 16, 32 |
| Compare match counter_1 | CMCNT_1 | R/W | H'0000 | H'FFFFCE0A | 8, 16 |
| Compare match constant register_1 | CMCOR_1 | R/W | H'FFFF | H'FFFFCE0C | 8, 16, 32 |

### 18.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether compare match counter (CMCNT) operates or is stopped.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | - | - | - | - | STR1 | STR0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 15 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | STR1 | 0 | R/W | Count Start 1<br>Specifies whether compare match counter 1 operates or is stopped.<br>0: CMCNT_1 count is stopped<br>1: CMCNT_1 count is started |
| 0 | STR0 | 0 | R/W | Count Start 0<br>Specifies whether compare match counter 0 operates or is stopped.<br>0: CMCNT_0 count is stopped<br>1: CMCNT_0 count is started |

### 18.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates compare match generation, enables interrupts and selects the counter input clock.

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

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 |
|---|---|---|---|---|
| 15 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 7 | CMF | 0 | (R/W)*[1] | Compare Match Flag<br>Indicates whether or not the values of CMCNT and CMCOR match.<br>0: CMCNT and CMCOR values do not match<br>[Clearing conditions]<br>• When 0 is written to this bit after reading CMF=1*[2]<br>• When CMT registers are accessed when the value of the DISEL bit of MRB in the DTC is 0 after activating the DTC by CMI interrupts.<br>[Setting condition]<br>1: CMCNT and CMCOR values match |
| 6 | CMIE | 0 | R/W | Compare Match Interrupt Enable<br>Enables or disables compare match interrupt (CMI) generation when CMCNT and CMCOR values match (CMF=1).<br>0: Compare match interrupt (CMI) disabled<br>1: Compare match interrupt (CMI) enabled |
| 5 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1, 0 | CKS[1:0] | 00 | R/W | Clock Select 1 and 0<br>Select the clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral operating clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS1 and CKS0.<br>00: Pϕ/8<br>01: Pϕ/32<br>10: Pϕ/128<br>11: Pϕ/512 |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. he flag is set by another compare match before writing 0 to the bit after reading it as 1, the flag will not be cleared by writing 0 to it once. In this case, read the bit as 1 again and write 0 to it.

### 18.2.3 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS1 and CKS0 in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock.

When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1.

The initial value of CMCNT 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/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

### 18.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

The initial value of CMCOR 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 |

## 18.3 Operation

### 18.3.1 Interval Count Operation

When an internal clock is selected with bits CKS1 and CKS0 in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1. When the CMIE bit in CMCSR is set to 1, a compare match interrupt (CMI) is requested. CMCNT then starts counting up again from H'0000.

Figure 18.2 shows the operation of the compare match counter.

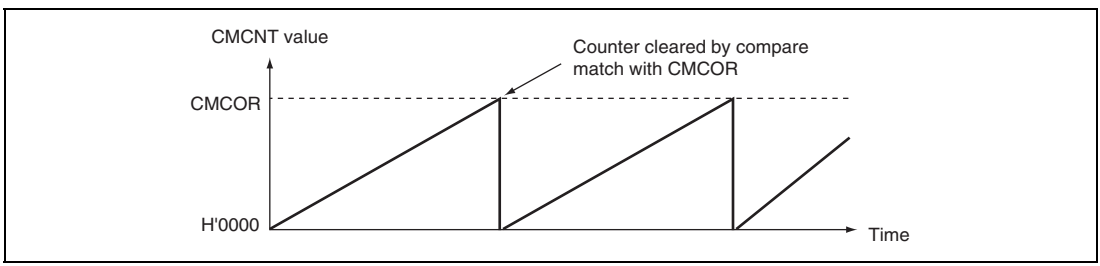



**Figure 18.2 Counter Operation**

### 18.3.2 CMCNT Count Timing

One of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the Pϕ clock can be selected with bits CKS1 and CKS0 in CMCSR. Figure 18.3 shows the timing.

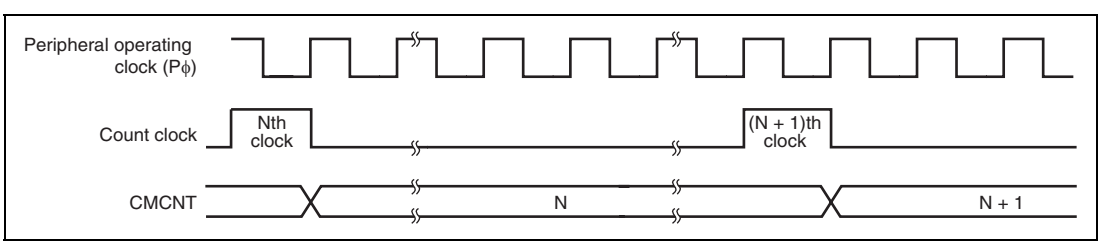



**Figure 18.3 Count Timing**

## 18.4 Interrupts

### 18.4.1 CMT Interrupt Sources and DTC Activation

The CMT has channels and each of them to which a different vector address is allocated has compare match interrupt. When both the interrupt request flag (CMF) and interrupt enable bit (CMIE) are set to 1, the corresponding interrupt request is output. When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 6, Interrupt Controller (INTC).

The data transfer controller (DTC) can be activated by an interrupt request. In this case, the priority between channels is fixed. See section 8, Data Transfer Controller (DTC), for details.

Table 18.2 lists the CMT interrupt sources.

**Table 18.2 Interrupt Source**

| Channel | Interrupt Source | Interrupt Enable Bit | Interrupt Flag Bit | DTC Activation | Priority |
|---|---|---|---|---|---|
| 0 | CMI_0 | CMIE | CMF | Possible | High |
| 1 | CMI_1 | CMIE | CMF | Possible | Low |

### 18.4.2 Timing of Setting Compare Match Flag

When CMCOR and CMCNT match, a compare match signal is generated and the CMF bit in CMCSR is set to 1. The compare match signal is generated in the last cycle in which the values match (when the CMCNT value is updated to H'0000). That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 18.4 shows the timing of CMF bit setting.

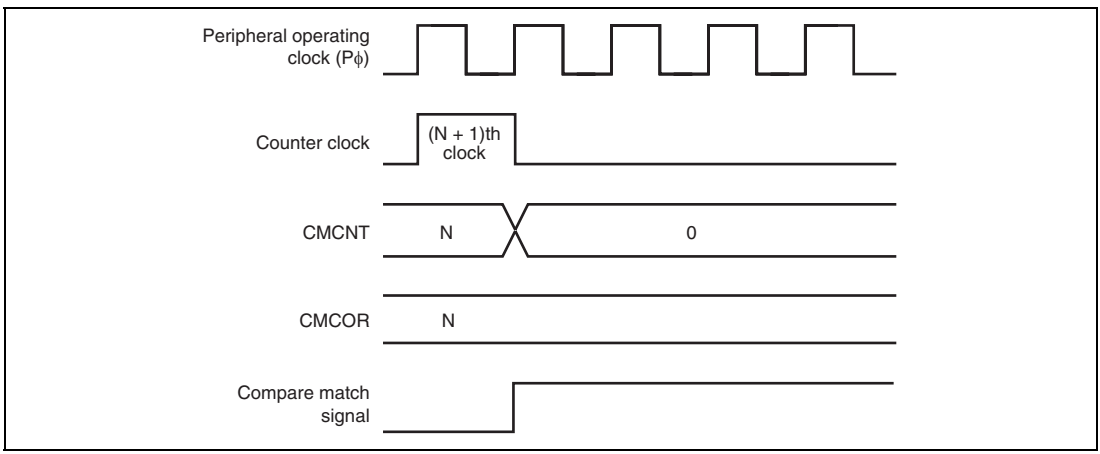



**Figure 18.4 Timing of CMF Setting**

### 18.4.3 Timing of Clearing Compare Match Flag

The CMF bit in CMCSR is cleared by reading 1 from this bit, then writing 0.

## 18.5 Usage Notes

### 18.5.1 Module Standby Mode Setting

The CMT operation can be disabled or enabled using the standby control register. The initial setting is for CMT operation to be halted. Access to a register is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes.

### 18.5.2 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated in the T2 cycle while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 18.5 shows the timing to clear the CMCNT counter.

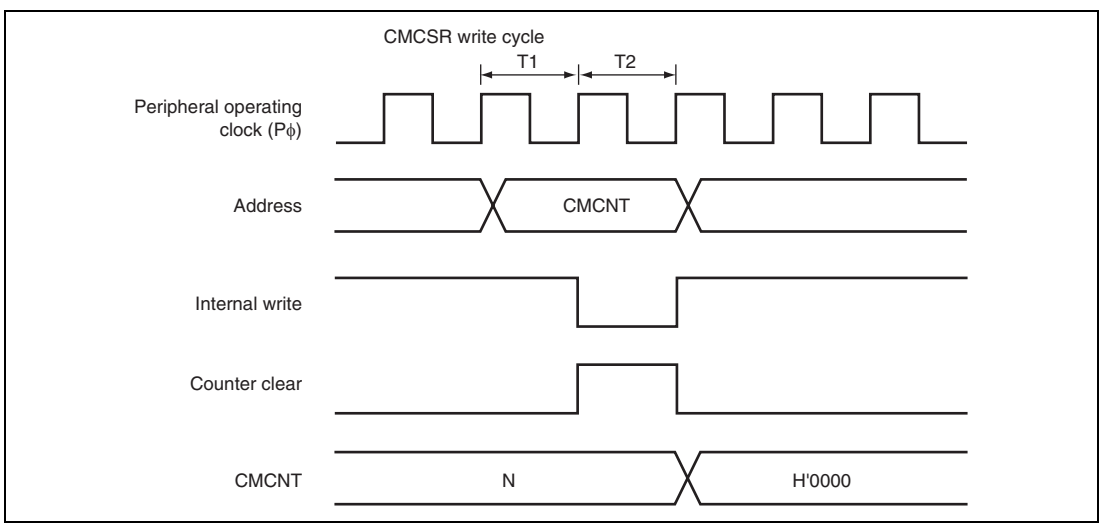



**Figure 18.5 Conflict between Write and Compare-Match Processes of CMCNT**

### 18.5.3 Conflict between Word-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in words, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 18.6 shows the timing to write to CMCNT in words.

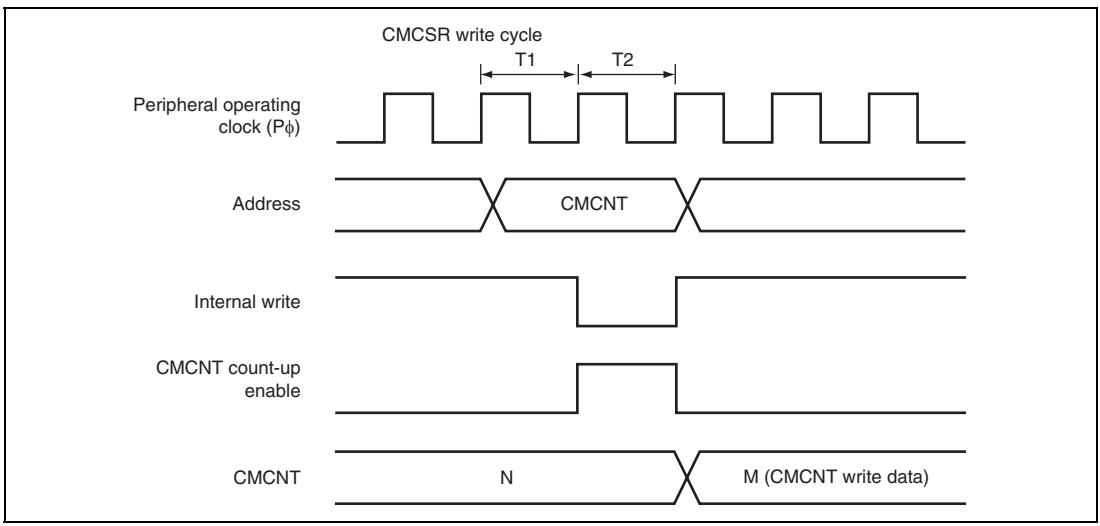



**Figure 18.6 Conflict between Word-Write and Count-Up Processes of CMCNT**

### 18.5.4 Conflict between Byte-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in bytes, the byte-writing has priority over the count-up. In this case, the count-up is not performed. The byte data on another side, which is not written to, is also not counted and the previous contents remain.

Figure 18.7 shows the timing when the count-up occurs in the T2 cycle while writing to CMCNT in bytes.

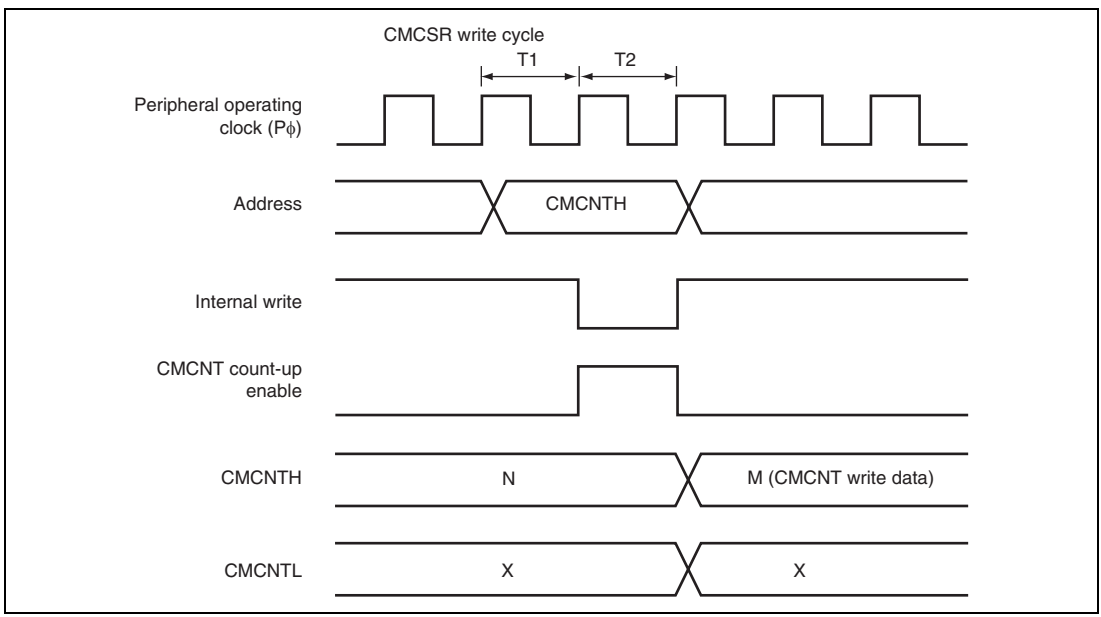



**Figure 18.7 Conflict between Byte-Write and Count-Up Processes of CMCNT**

### 18.5.5 Compare Match between CMCNT and CMCOR

Do not set the same value in CMCNT and CMCOR while CMCNT is not counting. If set, the CMF bit in CMCSR is set to 1 and CMCNT is cleared to H'0000.

# 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 Licence Specification, Robert Bosch GmbH, 1992
2. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
3. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
4. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
5. Road vehicles - Controller area network (CAN): Part 1: Data link layer and physical signalling (ISO-11898-1, 2003)

### 19.1.5 Features

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

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.

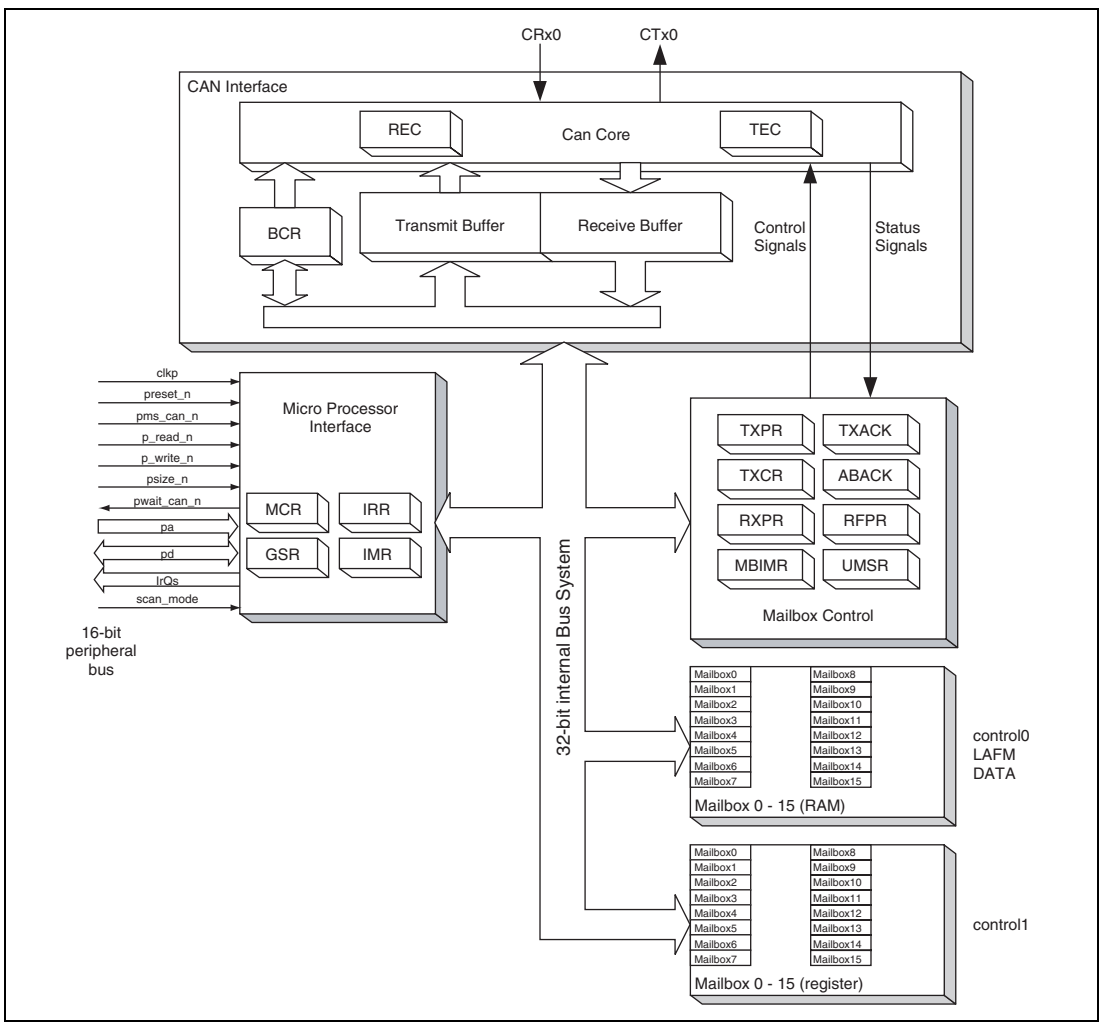



**Figure 19.1 RCAN-ET Architecture**

**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. In that case, LongWord (32-bit) access must be implemented as 2 consecutive word (16-bit) accesses. In this manual, LongWord access means the two consecutive accesses.

- Micro Processor Interface (MPI)

  The MPI allows communication between the Renesas CPU and RCAN-ET's registers/mailboxes to control the memory interface. It also contains the Wakeup Control logic that detects the CAN bus activities and notifies the MPI and the other parts of RCAN-ET so that the RCAN-ET can automatically exit the Sleep mode.

  It contains registers such as MCR, IRR, GSR and IMR.

- Mailbox

  The Mailboxes consists of RAM configured as message buffers and registers. There are 16 Mailboxes, and each mailbox has the following information.

  <RAM>

  — CAN message control (identifier, rtr, ide,etc)

  — CAN message data (for CAN Data frames)

  — Local Acceptance Filter Mask for reception

  <Registers>

  — CAN message control (dlc)

  — 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit

- Mailbox Control

  The Mailbox Control handles the following functions:

  — For received messages, compare the IDs and generate appropriate RAM addresses/data to store messages from the CAN Interface into the Mailbox and set/clear appropriate registers accordingly.

  — To transmit messages, RCAN-ET will run the internal arbitration to pick the correct priority message, and load the message from the Mailbox into the Tx-buffer of the CAN Interface and set/clear appropriate registers accordingly.

  — Arbitrates Mailbox accesses between the CPU and the Mailbox Control.

  — Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, UMSR and MBIMR.

- CAN Interface

  This block conforms to the requirements for a CAN Bus Data Link Controller which is specified in Ref. [2, 4]. It fulfils all the functions of a standard DLC as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and the logic which are specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Configuration Registers and various useful Test Modes. This block also contains functional entities to hold the data received and the data to be transmitted for the CAN Data Link Controller.

## 19.3 Programming Model – Overview

The purpose of this programming interface is to allow convenient, effective access to the CAN bus for efficient message transfer. Please bear in mind that the user manual reports all settings allowed by the RCAN-ET IP. Different use of RCAN-ET is not allowed.

### 19.3.1 Memory Map

The diagram of the memory map is shown below.

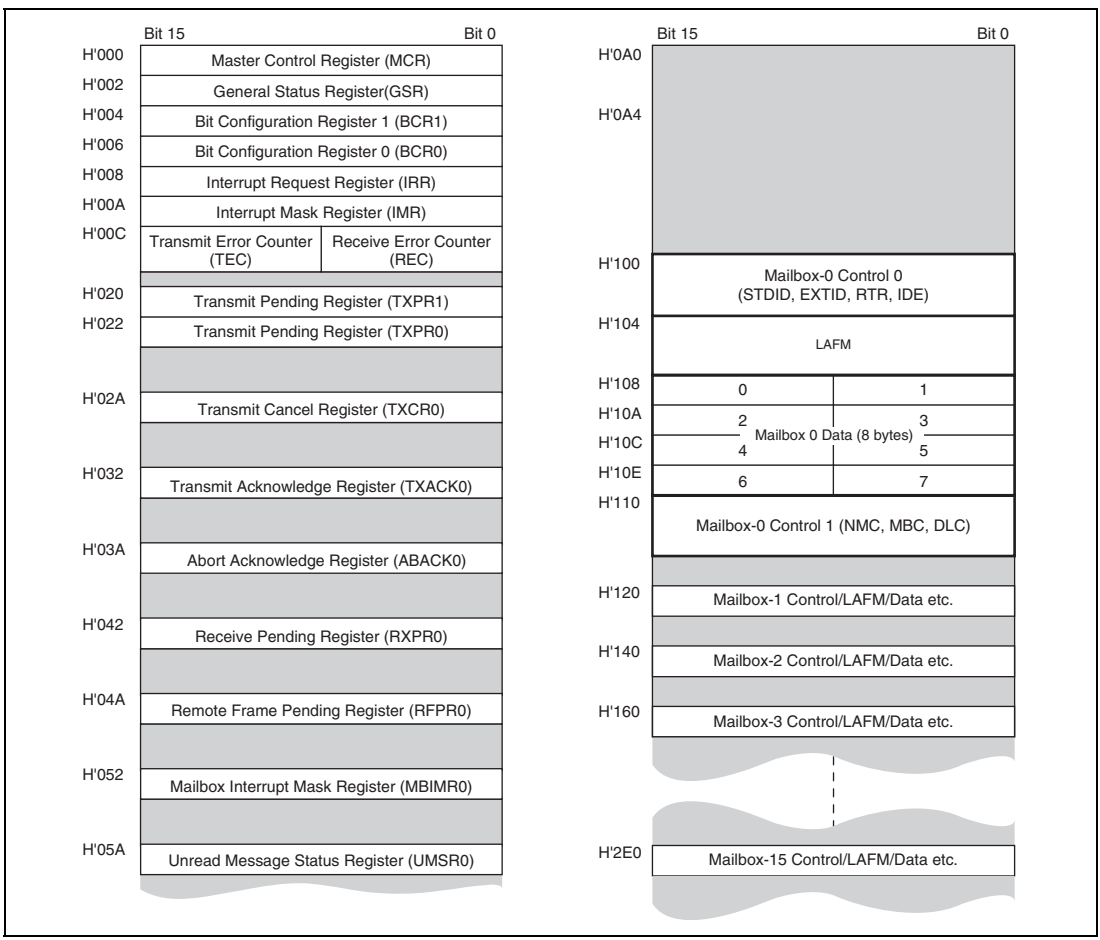



**Figure 19.2 RCAN-ET Memory Map**

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

### 19.3.2 Mailbox Structure

Mailboxes play a role as message buffers to transmit / receive CAN frames. Each Mailbox is comprised of 3 identical storage fields that are 1): Message Control, 2): Local Acceptance Filter Mask, 3): Message Data. The following table shows the address map for the control, LAFM, data and addresses for each mailbox.

| | Address | | | |
|---|---|---|---|---|
| | **Control0** | **LAFM** | **Data** | **Control1** |
| **Mailbox** | **4 bytes** | **4 bytes** | **8 bytes** | **2 bytes** |
| 0 (Receive Only) | 100 – 103 | 104– 107 | 108 – 10F | 110 – 111 |
| 1 | 120 – 123 | 124 – 127 | 128 – 12F | 130 – 131 |
| 2 | 140 – 143 | 144 – 147 | 148 – 14F | 150 – 151 |
| 3 | 160 – 163 | 164 - 167 | 168 – 16F | 170 – 171 |
| 4 | 180 – 183 | 184 – 187 | 188 – 18F | 190 – 191 |
| 5 | 1A0 – 1A3 | 1A4 – 1A7 | 1A8 – 1AF | 1B0 – 1B1 |
| 6 | 1C0 – 1C3 | 1C4 – 1C7 | 1C8 – 1CF | 1D0 – 1D1 |
| 7 | 1E0 – 1E3 | 1E4 – 1E7 | 1E8 – 1EF | 1F0 – 1F1 |
| 8 | 200 – 203 | 204 – 207 | 208 – 20F | 210 – 211 |
| 9 | 220 – 223 | 224 – 227 | 228 – 22F | 230 – 231 |
| 10 | 240 – 243 | 244 – 247 | 248 – 24F | 250 – 251 |
| 11 | 260 – 263 | 264 – 267 | 268 – 26F | 270 – 271 |
| 12 | 280 – 283 | 284 – 287 | 288 – 28F | 290 – 291 |
| 13 | 2A0 – 2A3 | 2A4 – 2A7 | 2A8 – 2AF | 2B0 – 2B1 |
| 14 | 2C0 – 2C3 | 2C4 – 2C7 | 2C8 – 2CF | 2D0 – 2D1 |
| 15 | 2E0 – 2E3 | 2E4 – 2E7 | 2E8 – 2EF | 2F0 – 2F1 |

Mailbox-0 is a receive-only box, and all the other Mailboxes can operate as both receive and transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message Control. The following diagram shows the structure of a Mailbox in detail.

**Table 19.1 Roles of Mailboxes**

| | Tx | Rx |
|---|---|---|
| MB15-1 | OK | OK |
| MB0 | — | OK |

MB0 (reception MB) — Byte: 8-bit access, Word: 16-bit access, LW (LongWord): 32-bit access

| Address | Data Bus 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Access Size | Field 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 |
| H'106 + N*32 | EXTID_LAFM[15:0] | | | | | | | | | | | | | | | | Word | |
| H'108 + N*32 | MSG_DATA_0 (first Rx/Tx Byte) | | | | | | | | MSG_DATA_1 | | | | | | | | Byte/Word/LW | Data |
| H'10A + N*32 | MSG_DATA_2 | | | | | | | | MSG_DATA_3 | | | | | | | | Byte/Word | |
| H'10C + N*32 | MSG_DATA_4 | | | | | | | | MSG_DATA_5 | | | | | | | | Byte/Word/LW | |
| H'10E + N*32 | MSG_DATA_6 | | | | | | | | MSG_DATA_7 | | | | | | | | Byte/Word | |
| H'110 + N*32 | 0 | 0 | NMC | 0 | 0 | MBC[2:0] | | | 0 | 0 | 0 | 0 | DLC[3:0] | | | | Byte/Word | Control 1 |

MBC[1] is fixed to "1"

MB15-1 (MB for transmission/reception)

| Address | Data Bus 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Access Size | Field 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 |
| H'106 + N*32 | EXTID_LAFM[15:0] | | | | | | | | | | | | | | | | Word | |
| H'108 + N*32 | MSG_DATA_0 (first Rx/Tx Byte) | | | | | | | | MSG_DATA_1 | | | | | | | | Byte/Word/LW | Data |
| H'10A + N*32 | MSG_DATA_2 | | | | | | | | MSG_DATA_3 | | | | | | | | Byte/Word | |
| H'10C + N*32 | MSG_DATA_4 | | | | | | | | MSG_DATA_5 | | | | | | | | Byte/Word/LW | |
| H'10E + N*32 | MSG_DATA_6 | | | | | | | | MSG_DATA_7 | | | | | | | | Byte/Word | |
| H'110 + N*32 | 0 | 0 | NMC | ATX | DART | MBC[2:0] | | | 0 | 0 | 0 | 0 | DLC[3:0] | | | | Byte/Word | Control 1 |

**Figure 19.3 Mailbox-N Structure**

Notes: 1. All bits shadowed in grey are reserved and must be written LOW. The value returned by a read may not always be ‘0’ and should not be relied upon.

2. ATX and DART are not supported by Mailbox-0, and the MBC setting of Mailbox-0 is limited.

3. ID Reorder (MCR15) can change the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

### (1) Message Control Field

**STDID[10:0]**: These bits set the identifier (standard identifier) of data frames and remote frames.

**EXTID[17:0]**: These bits set the identifier (extended identifier) of data frames and remote frames.

**RTR** (Remote Transmission Request bit) : Used to distinguish between data frames and remote frames. This bit is overwritten by received CAN Frames depending on Data Frames or Remote Frames.

**Important:** Please note that, when ATX bit is set with the setting MBC=001(bin), the RTR bit will never be set. When a Remote Frame is received, the CPU can be notified by the corresponding RFPR set or IRR[2] (Remote Frame Request Interrupt), however, as RCAN-ET needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

**Important:** In order to support automatic answer to remote frame when MBC=001(bin) is used and ATX=1 the RTR flag must be programmed to zero to allow data frame to be transmitted.

Note: when a Mailbox is configured to send a remote frame request the DLC used for transmission is the one stored into the Mailbox.

| RTR | Description |
|---|---|
| 0 | Data frame |
| 1 | Remote frame |

**IDE** (Identifier Extension bit) : Used to distinguish between the standard format and extended format of CAN data frames and remote frames.

| IDE | Description |
|---|---|
| 0 | Standard format |
| 1 | Extended format |

- Mailbox-0

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

Note: MBC[1] of MB0 is always "1".

- Mailbox-15 to 1

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | NMC | ATX | DART | MBC[2:0] | | | 0 | 0 | 0 | 0 | DLC[3:0] | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W |

**NMC (New Message Control):** When this bit is set to '0', the Mailbox of which the RXPR or RFPR bit is already set does not store the new message but maintains the old one and sets the UMSR correspondent bit. When this bit is set to '1', the Mailbox of which the RXPR or RFPR bit is already set overwrites with the new message and sets the UMSR correspondent bit.

**Important:** Please note that if a remote frame is overwritten with a data frame or vice versa could be that both RXPR and RFPR flags (together with UMSR) are set for the same Mailbox. In this case the RTR bit within the Mailbox Control Field should be relied upon.

| NMC | Description |
|---|---|
| 0 | Overrun mode (Initial value) |
| 1 | Overwrite mode |

**ATX (Automatic Transmission of Data Frame):** When this bit is set to '1' and a Remote Frame is received into the Mailbox DLC is stored. Then, a Data Frame is transmitted from the same Mailbox using the current contents of the message data and updated DLC by setting the corresponding TXPR automatically. The scheduling of transmission is still governed by ID priority or Mailbox priority as configured with the Message Transmission Priority control bit (MCR.2). In order to use this function, MBC[2:0] needs to be programmed to be '001' (Bin). When a transmission is performed by this function, the DLC (Data Length Code) to be used is the one that has been received. Application needs to guarantee that the DLC of the remote frame correspond to the DLC of the data frame requested.

**Important:** When ATX is used and MBC=001 (Bin) the filter for the IDE bit cannot be used since ID of remote frame has to be exactly the same as that of data frame as the reply message.

**Important:** Please note that, when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the CPU will be notified by the corresponding RFPR set, however, as RCAN-ET needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

**Important:** Please note that in case of overrun condition (UMSR flag set when the Mailbox has its NMC = 0) the message received is discarded. In case a remote frame is causing overrun into a Mailbox configured with ATX = 1, the transmission of the corresponding data frame may be triggered only if the related RFPR flag is cleared by the CPU when the UMSR flag is set. In such case RFPR flag would get set again.

| ATX | Description |
|---|---|
| 0 | Automatic Transmission of Data Frame disabled (Initial value) |
| 1 | Automatic Transmission of Data Frame enabled |

**DART (Disable Automatic Re-Transmission):** When this bit is set, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is set to '0', RCAN-ET tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.

| DART | Description |
|---|---|
| 0 | Re-transmission enabled (Initial value) |
| 1 | Re-Transmission disabled |

**MBC[2:0] (Mailbox Configuration):** These bits configure the nature of each Mailbox as follows. When MBC=111 (Bin), the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. The MBC='110', '101' and '100' settings are prohibited. When the MBC is set to any other value, the LAFM field becomes available. Please don't set TXPR when MBC is set as reception. There is no hardware protection, and TXPR remains set. MBC[1] of Mailbox-0 is fixed to "1" by hardware. This is to ensure that MB0 cannot be configured to transmit Messages.

| MBC[2] | MBC[1] | MBC[0] | Data Frame Transmit | Remote Frame Transmit | Data Frame Receive | Remote Frame Receive | Remarks |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | Yes | Yes | No | No | • Not allowed for Mailbox-0 |
| 0 | 0 | 1 | Yes | Yes | No | Yes | • Can be used with ATX*<br>• Not allowed for Mailbox-0<br>• LAFM can be used |
| 0 | 1 | 0 | No | No | Yes | Yes | • Allowed for Mailbox-0<br>• LAFM can be used |
| 0 | 1 | 1 | No | No | Yes | No | • Allowed for Mailbox-0<br>• LAFM can be used |
| 1 | 0 | 0 | Setting prohibited | | | | |
| 1 | 0 | 1 | Setting prohibited | | | | |
| 1 | 1 | 0 | Setting prohibited | | | | |
| 1 | 1 | 1 | Mailbox inactive (Initial value) | | | | |

Notes: * In order to support automatic retransmission, RTR shall be "0" when MBC=001(bin) and ATX=1.
When ATX=1 is used the filter for IDE must not be used

**DLC[3:0] (Data Length Code):** These bits encode the number of data bytes from 0,1, 2, … 8 that will be transmitted in a data frame. Please note that when a remote frame request is transmitted the DLC value to be used must be the same as the DLC of the data frame that is requested.

| DLC[3] | DLC[2] | DLC[1] | DLC[0] | Description |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Data Length = 0 bytes (Initial value) |
| 0 | 0 | 0 | 1 | Data Length = 1 byte |
| 0 | 0 | 1 | 0 | Data Length = 2 bytes |
| 0 | 0 | 1 | 1 | Data Length = 3 bytes |
| 0 | 1 | 0 | 0 | Data Length = 4 bytes |
| 0 | 1 | 0 | 1 | Data Length = 5 bytes |
| 0 | 1 | 1 | 0 | Data Length = 6 bytes |
| 0 | 1 | 1 | 1 | Data Length = 7 bytes |
| 1 | x | x | x | Data Length = 8 bytes |

### (2) Local Acceptance Filter Mask (LAFM)

This area is used as Local Acceptance Filter Mask (LAFM) for receive boxes**.**

**LAFM:** When MBC is set to 001, 010, 011 (Bin), this field is used as LAFM Field. It allows a Mailbox to accept more than one identifier. The LAFM is comprised of two 16-bit read/write areas as follows.

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | |

**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" |

### (3) Message Data Fields

Storage for the CAN message data that is transmitted or received. MSG_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 through to bit 0.

### 19.3.3 RCAN-ET Control Registers

The following sections describe RCAN-ET control registers. The address is mapped as follow.

**Important:** These registers can only be accessed in Word size (16-bit).

| Description | Address | Name | Access Size (bits) |
|---|---|---|---|
| Master Control Register | 000 | MCR | Word |
| General Status Register | 002 | GSR | Word |
| Bit Configuration Register 1 | 004 | BCR1 | Word |
| Bit Configuration Register 0 | 006 | BCR0 | Word |
| Interrupt Request Register | 008 | IRR | Word |
| Interrupt Mask Register | 00A | IMR | Word |
| Error Counter Register | 00C | TEC/REC | Word |

**Figure 19.5 RCAN-ET Control Registers**

#### (1) Master Control Register (MCR)

The Master Control Register (MCR) is a 16-bit read/write register that controls RCAN-ET.

- MCR (Address = H'000)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MCR15 | MCR14 | - | - | - | TST[2:0] | | | MCR7 | MCR6 | MCR5 | - | - | MCR2 | MCR1 | MCR0 |
| Initial value: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W: | R/W | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W | R/W |

**Bit 15 — ID Reorder (MCR15):** This bit changes the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

| Bit15 : MCR15 | Description |
|---|---|
| 0 | RCAN-ET is the same as HCAN2 |
| 1 | RCAN-ET is not the same as HCAN2 (Initial value) |

MCR15 (ID Reorder) = 0

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H'100 + N*32 | IDE | RTR | 0 | STDID[10:0] | | | | | | | | | | | EXTID[17:16] | | Word/LW | Control 0 |
| H'102 + N*32 | EXTID[15:0] | | | | | | | | | | | | | | | | Word | |
| H'104 + N*32 | IDE_ LAFM | 0 | 0 | STDID_LAFM[10:0] | | | | | | | | | | | EXTID_LAFM [17:16] | | Word/LW | LAFM Field |
| H'106 + N*32 | EXTID_LAFM[15:0] | | | | | | | | | | | | | | | | Word | |

**Figure 19.6 ID Reorder**

This bit can be modified only in reset mode.

**Bit 14 — Auto Halt Bus Off (MCR14):** If both this bit and MCR6 are set, MCR1 is automatically set as soon as RCAN-ET enters BusOff.

| Bit14 : MCR14 | Description |
|---|---|
| 0 | RCAN-ET remains in BusOff for normal recovery sequence (128 × 11 Recessive Bits) (Initial value) |
| 1 | RCAN-ET moves directly into Halt Mode after it enters BusOff if MCR6 is set. |

This bit can be modified only in reset mode.

**Bit 13 — Reserved**. The written value should always be '0' and the returned value is '0'.

**Bit 12 — Reserved**. The written value should always be '0' and the returned value is '0'.

**Bit 11 — Reserved**. The written value should always be '0' and the returned value is '0'.

**Bits 10 to 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 19.4.1, Test Mode Settings.

Please note that the test modes are allowed only for diagnosis and tests and not when RCAN-ET is used in normal operation.

| Bit10: TST2 | Bit9: TST1 | Bit8: TST0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | Normal Mode (initial value) |
| 0 | 0 | 1 | Listen-Only Mode (Receive-Only Mode) |
| 0 | 1 | 0 | Self Test Mode 1 (External) |
| 0 | 1 | 1 | Self Test Mode 2 (Internal) |
| 1 | 0 | 0 | Write Error Counter |
| 1 | 0 | 1 | Error Passive Mode |
| 1 | 1 | 0 | setting prohibited |
| 1 | 1 | 1 | setting prohibited |

**Bit 7 — Auto-wake Mode (MCR7):** MCR7 enables or disables the Auto-wake mode. If this bit is set, the RCAN-ET automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared the RCAN-ET does not automatically cancel the sleep mode.

RCAN-ET cannot store the message that wakes it up.

Note: MCR7 cannot be modified while in sleep mode.

| Bit7 : MCR7 | Description |
|---|---|
| 0 | Auto-wake by CAN bus activity disabled (Initial value) |
| 1 | Auto-wake by CAN bus activity enabled |

**Bit 6 — Halt during Bus Off (MCR6):** MCR6 enables or disables entering Halt mode immediately when MCR1 is set during Bus Off. This bit can be modified only in Reset or Halt mode. Please note that when Halt is entered in Bus Off the CAN engine is also recovering immediately to Error Active mode.

| Bit6 : MCR6 | Description |
|---|---|
| 0 | If MCR[1] is set, RCAN-ET will not enter Halt mode during Bus Off but wait up to end of recovery sequence (Initial value) |
| 1 | Enter Halt mode immediately during Bus Off if MCR[1] or MCR[14] are asserted. |

**Bit 5 — Sleep Mode (MCR5):** Enables or disables Sleep mode transition. If this bit is set, while RCAN-ET is in halt mode, the transition to sleep mode is enabled. Setting MCR5 is allowed after entering Halt mode. The two Error Counters (REC, TEC) will remain the same during Sleep mode. This mode will be exited in two ways:

1. by writing a '0' to this bit position.
2. or, if MCR[7] is enabled, after detecting a dominant bit on the CAN bus.

If Auto wake up mode is disabled, RCAN-ET will ignore all CAN bus activities until the sleep mode is terminated. When leaving this mode the RCAN-ET will synchronise to the CAN bus (by checking for 11 recessive bits) before joining CAN Bus activity. This means that, when the No.2 method is used, RCAN-ET will miss the first message to receive. CAN transceivers stand-by mode will also be unable to cope with the first message when exiting stand by mode, and the S/W needs to be designed in this manner.

In sleep mode only the following registers can be accessed: MCR, GSR, IRR and IMR.

**Important:** RCAN-ET is required to be in Halt mode before requesting to enter in Sleep mode. That allows the CPU to clear all pending interrupts before entering sleep mode. Once all interrupts are cleared RCAN-ET must leave the Halt mode and enter Sleep mode simultaneously (by writing MCR[5]=1 and MCR[1]=0 at the same time).

| Bit 5 : MCR5 | Description |
|---|---|
| 0 | RCAN-ET sleep mode released (Initial value) |
| 1 | Transition to RCAN-ET sleep mode enabled |

**Bit 4 — Reserved**. The written value should always be '0' and the returned value is '0'.

**Bit 3 — Reserved**. The written value should always be '0' and the returned value is '0'.

**Bit 2 — Message Transmission Priority (MCR2):** MCR2 selects the order of transmission for pending transmit data. If this bit is set, pending transmit data are sent in order of the bit position in the Transmission Pending Register (TXPR). The order of transmission starts from Mailbox-15 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission).

If MCR2 is cleared, all messages for transmission are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field (STDID + IDE bit + EXTID (if IDE=1) + RTR bit) with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit (internal arbitration works in the same way as the arbitration on the CAN Bus between two CAN nodes starting transmission at the same time).

This bit can be modified only in Reset or Halt mode.

| Bit 2 : MCR2 | Description |
|---|---|
| 0 | Transmission order determined by message identifier priority (Initial value) |
| 1 | Transmission order determined by mailbox number priority (Mailbox-15 → Mailbox-1) |

**Bit 1 — Halt Request (MCR1):** Setting the MCR1 bit causes the CAN controller to complete its current operation and then enter Halt mode (where it is cut off from the CAN bus). The RCAN-ET remains in Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interface does not join the CAN bus activity and does not store messages or transmit messages. All the user registers (including Mailbox contents and TEC/REC) remain unchanged with the exception of IRR0 and GSR4 which are used to notify the halt status itself. If the CAN bus is in idle or intermission state regardless of MCR6, RCAN-ET will enter Halt Mode within one Bit Time. If MCR6 is set, a halt request during Bus Off will be also processed within one Bit Time. Otherwise the full Bus Off recovery sequence will be performed beforehand. Entering the Halt Mode can be notified by IRR0 and GSR4.

If both MCR14 and MCR6 are set, MCR1 is automatically set as soon as RCAN-ET enters BusOff.

In the Halt mode, the RCAN-ET configuration can be modified with the exception of the Bit Timing setting, as it does not join the bus activity. MCR[1] has to be cleared by writing a '0' in order to re-join the CAN bus. After this bit has been cleared, RCAN-ET waits until it detects 11 recessive bits, and then joins the CAN bus.

Note: After issuing a Halt request the CPU is not allowed to set TXPR or TXCR or clear MCR1 until the transition to Halt mode is completed (notified by IRR0 and GSR4). After MCR1 is set this can be cleared only after entering Halt mode or through a reset operation (SW or HW).

Note: Transition into or recovery from HALT mode, is only possible if the BCR1 and BCR0 registers are configured to a proper Baud Rate.

| Bit 1 : MCR1 | Description |
|---|---|
| 0 | Clear Halt request (Initial value) |
| 1 | Halt mode transition request |

**Bit 0 — Reset Request (MCR0):** Controls resetting of the RCAN-ET module. When this bit is changed from '0' to '1' the RCAN-ET controller enters its reset routine, re-initialising the internal logic, which then sets GSR3 and IRR0 to notify the reset mode. During a re-initialisation, all user registers are initialised.

RCAN-ET can be re-configured while this bit is set. This bit has to be cleared by writing a '0' to join the CAN bus. After this bit is cleared, the RCAN-ET module waits until it detects 11 recessive bits, and then joins the CAN bus. The Baud Rate needs to be set up to a proper value in order to sample the value on the CAN Bus.

After Power On Reset, this bit and GSR3 are always set. This means that a reset request has been made and RCAN-ET needs to be configured.

The Reset Request is equivalent to a Power On Reset but controlled by Software.

| Bit 0 : MCR0 | Description |
|---|---|
| 0 | Clear Reset Request |
| 1 | CAN Interface reset mode transition request (Initial value) |

### (2) General Status Register (GSR)

The General Status Register (GSR) is a 16-bit read-only register that indicates the status of RCAN-ET.

- GSR (Address = H'002)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | GSR5 | GSR4 | GSR3 | GSR2 | GSR1 | GSR0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

**Bits 15 to 6: Reserved**. The written value should always be '0' and the returned value is '0'.

**Bit 5 — Error Passive Status Bit (GSR5):** Indicates whether the CAN Interface is in Error Passive or not. This bit will be set high as soon as the RCAN-ET enters the Error Passive state and is cleared when the module enters again the Error Active state (this means the GSR5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR5 and GSR0 must be considered.

| Bit 5 : GSR5 | Description |
|---|---|
| 0 | RCAN-ET is not in Error Passive or in Bus Off status (Initial value)<br>[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)<br>[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)<br>[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)<br>[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 7$^{th}$ bit of End Of Frame. GSR2 is set at the 3$^{rd}$ 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 9$^{th}$ bit is equivalent to GSR0.

### (3) Bit Configuration Register (BCR0, BCR1)

The bit configuration registers (BCR0 and BCR1) are 2 X 16-bit read/write register that are used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN Interface.

The Time quanta is defined as:

$$Timequanta = \frac{2 * BRP}{f_{clk}}$$

Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral bus frequency.

- BCR1 (Address = H'004)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TSG1[3:0] | | | | - | TSG2[2:0] | | | - | - | SJW[1:0] | | - | - | - | BSP |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R | R/W | R/W | R | R | R | R/W |

Please refer to the table on section 0 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 |
| :<br>: | :<br>: | :<br>: | :<br>: | :<br>: |
| 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 sec. 0 |
| 0 | 1 | 0 | PHSEG2 = 3 time quanta |
| 0 | 1 | 1 | PHSEG2 = 4 time quanta |
| 1 | 0 | 0 | PHSEG2 = 5 time quanta |
| 1 | 0 | 1 | PHSEG2 = 6 time quanta |
| 1 | 1 | 0 | PHSEG2 = 7 time quanta |
| 1 | 1 | 1 | PHSEG2 = 8 time quanta |

**Bits 7 and 6: Reserved**. The written value should always be '0' and the returned value is '0'.

**Bits 5 and 4 — ReSynchronisation Jump Width (SJW[1:0] = BCR0[5:4]):** These bits set the synchronisation jump width.

| Bit 5: SJW[1] | Bit 4: SJW[0] | Description |
|---|---|---|
| 0 | 0 | Synchronisation Jump width = 1 time quantum (Initial value) |
| 0 | 1 | Synchronisation Jump width = 2 time quanta |
| 1 | 0 | Synchronisation Jump width = 3 time quanta |
| 1 | 1 | Synchronisation Jump width = 4 time quanta |

**Bits 3 to 1: Reserved**. The written value should always be '0' and the returned value is '0'.

**Bit 0 — Bit Sample Point (BSP = BCR1[0]):** Sets the point at which data is sampled.

| Bit 0 : BSP | Description |
|---|---|
| 0 | Bit sampling at one point (end of time segment 1) (Initial value) |
| 1 | Bit sampling at three points (rising edge of the last three clock cycles of PHSEG1) |

- BCR0 (Address = H'006)

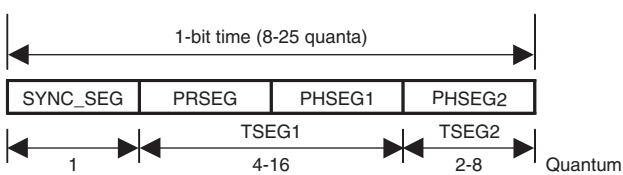
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | BRP[7:0] | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: |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

**Bits 8 to 15: Reserved**. The written value should always be '0' and the returned value is '0'.

**Bits 7 to 0 — Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]):** These bits are used to define the peripheral bus clock periods contained in a Time Quantum.

| Bit 7: BRP[7] | Bit 6: BRP[6] | Bit 5: BRP[5] | Bit 4: BRP[4] | Bit 3: BRP[3] | Bit 2: BRP[2] | Bit 1: BRP[1] | Bit 0: BRP[0] | Description |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 X peripheral bus clock (Initial value) |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 X peripheral bus clock |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 6 X peripheral bus clock |
| :<br>: | :<br>: | :<br>: | :<br>: | :<br>: | :<br>: | :<br>: | :<br>: | 2*(register value+1) X peripheral bus clock |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 512 X peripheral bus clock |

- Requirements of Bit Configuration Register




SYNC_SEG: Segment for establishing synchronisation of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks.

PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronisation (resynchronisation) is established.)

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

TSEG1: TSG1 + 1

TSEG2: TSG2 + 1

The RCAN-ET Bit Rate Calculation is:

$$\text{Bit Rate} = \frac{f_{clk}}{2 * (BRP + 1) * (TSEG1 + TSEG2 + 1)}$$

where BRP is given by the register value and TSEG1 and TSEG2 are derived values from TSG1 and TSG2 register values. The '+ 1' in the above formula is for the Sync-Seg which duration is 1 time quanta.

$f_{CLK}$ = Peripheral Clock

BCR Setting Constraints

TSEG1min > TSEG2 ≥ SJWmax (SJW = 1 to 4)

8 ≤ TSEG1 + TSEG2 + 1 ≤ 25 time quanta (TSEG1 + TSEG2 + 1 = 7 is not allowed)

TSEG2 ≥ 2

These constraints allow the setting range shown in the table below for TSEG1 and TSEG2 in the Bit Configuration Register. The number in the table shows possible setting of SJW. "No" shows that there is no allowed combination of TSEG1 and TSEG2.

| | | 001 | 010 | 011 | 100 | 101 | 110 | 111 | TSG2 |
|---|---|---|---|---|---|---|---|---|---|
| | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | TSEG2 |
| **TSG1** | **TSEG1** | | | | | | | | |
| **0011** | **4** | No | 1-3 | No | No | No | No | No | |
| **0100** | **5** | 1-2 | 1-3 | 1-4 | No | No | No | No | |
| **0101** | **6** | 1-2 | 1-3 | 1-4 | 1-4 | No | No | No | |
| **0110** | **7** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | No | No | |
| **0111** | **8** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | No | |
| **1000** | **9** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | |
| **1001** | **10** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | |
| **1010** | **11** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | |
| **1011** | **12** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | |
| **1100** | **13** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | |
| **1101** | **14** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | |
| **1110** | **15** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | |
| **1111** | **16** | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | |

Example 1: To have a Bit rate of 500 Kbps with a frequency of fclk = 40 MHz it is possible to set: BRP = 43, TSEG1 = 6, TSEG2 = 3.

Then the configuration to write is BCR1 = 5200 and BCR0 = 0003.

Example 2: To have a Bit rate of 250 Kps with a frequency of 35 MHz it is possible to set: BPR = 4, TSEG1 = 8, TSEG2 = 5.

Then the configuration to write is BCR1 = 7400 and BCR0 = 0004.

### (4) Interrupt Request Register (IRR)

The interrupt register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

- IRR (Address = H'008)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | IRR13 | IRR12 | - | - | IRR9 | IRR8 | IRR7 | IRR6 | IRR5 | IRR4 | IRR3 | IRR2 | IRR1 | IRR0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W: | R | R | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R | R | R/W |

**Bits 15 and 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)<br>[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)<br>[Clearing condition] Writing 1 |
| 1 | [Setting condition] dominant bit level detection on the Rx line while in sleep mode |

**Bits 11 to 10: Reserved**

**Bit 9 — Message Overrun/Overwrite Interrupt Flag (IRR9):** Flag indicating that a message has been received but the existing message in the matching Mailbox has not been read as the corresponding RXPR or RFPR is already set to '1' and not yet cleared by the CPU. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (New Message Control) bit. This bit is cleared when all bit in UMSR (Unread Message Status Register) are cleared (by writing '1') or by setting MBIMR (MailBox interrupt Mast Register) for all UMSR flag set . It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

| Bit 9: IRR9 | Description |
|---|---|
| 0 | No pending notification of message overrun/overwrite<br>[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<br>[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)<br>[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<br>[Setting condition]<br>When one of the TXPR bits is cleared by completion of transmission or completion of transmission abort, i.e., when a TXACK or ABACK bit is set (if MBIMR=0). |

**Bit 7 — Overload Frame (IRR7):** Flag indicating that the RCAN-ET has detected a condition that should initiate the transmission of an overload frame. Note that on the condition of transmission being prevented, such as listen only mode, an Overload Frame will NOT be transmitted, but IRR7 will still be set. IRR7 remains asserted until reset by writing a '1' to this bit position - writing a '0' has no effect.

| Bit 7: IRR7 | Description |
|---|---|
| 0 | [Clearing condition] Writing 1 (Initial value) |
| 1 | [Setting conditions] Overload condition detected |

**Bit 6 — Bus Off Interrupt Flag (IRR6):** This bit is set when RCAN-ET enters the Bus-off state or when RCAN-ET leaves Bus-off and returns to Error-Active. The cause therefore is the existing condition TEC ≥ 256 at the node or the end of the Bus-off recovery sequence (128X11 consecutive recessive bits) or the transition from Bus Off to Halt (automatic or manual). This bit remains set even if the RCAN-ET node leaves the bus-off condition, and needs to be explicitly cleared by S/W. The S/W is expected to read the GSR0 to judge whether RCAN-ET is in the bus-off or error active status. It is cleared by writing a '1' to this bit position even if the node is still bus-off. Writing a '0' has no effect.

| Bit 6: IRR6 | Description |
|---|---|
| 0 | [Clearing condition] Writing 1 (Initial value) |
| 1 | Enter Bus off state caused by transmit error or Error Active state returning from Bus-off<br>[Setting condition] When TEC becomes ≥ 256 or End of Bus-off after 128X11 consecutive recessive bits or transition from Bus Off to Halt |

**Bit 5 — Error Passive Interrupt Flag (IRR5):** Interrupt flag indicating the error passive state caused by the transmit or receive error counter or by Error Passive forced by test mode. This bit is reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the node may still be error passive. Please note that the SW needs to check GSR0 and GSR5 to judge whether RCAN-ET is in Error Passive or Bus Off status.

| Bit 5: IRR5 | Description |
|---|---|
| 0 | [Clearing condition] Writing 1 (Initial value) |
| 1 | Error passive state caused by transmit/receive error<br>[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<br>[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<br>[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<br>[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<br>[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 19.9.

IRR0 is set by the transition from "0" to "1" of GSR3 or GSR4 or by transition from Halt mode to Sleep mode. So, IRR0 is not set if RCAN-ET enters Halt mode again right after exiting from Halt mode, without GSR4 being cleared. Similarly, IRR0 is not set by direct transition from Sleep mode to Halt Request. At the transition from Halt/Sleep mode to Transition/Reception, clearing GSR4 needs (one-bit time - TSEG2) to (one-bit time * 2 - TSEG2).

In the case of Reset mode, IRR0 is set, however, the interrupt to the CPU is not asserted since IMR0 is automatically set by initialization.

| 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)<br>[Setting condition] When reset/halt/sleep transition is completed after a reset (MCR0 or HW) or Halt mode (MCR1) or Sleep mode (MCR5) is requested |

### (5) Interrupt Mask Register (IMR)

The interrupt mask register is a 16 bit register that protects all corresponding interrupts in the Interrupt Request Register (IRR) from generating an output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to '1'. This register can be read or written at any time. The IMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the IRR.

- IMR (Address = H'00A)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IMR15 | IMR14 | IMR13 | IMR12 | IMR11 | IMR10 | IMR9 | IMR8 | IMR7 | IMR6 | IMR5 | IMR4 | IMR3 | IMR2 | IMR1 | IMR0 |
| Initial value: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

**Bits 15 to 0:** Maskable interrupt sources corresponding to IRR[15:0] respectively. When a bit is set, the interrupt signal is not generated, although setting the corresponding IRR bit is still performed.

| Bit[15:0]: IMRn | Description |
|---|---|
| 0 | Corresponding IRR is not masked (IRQ is generated for interrupt conditions) |
| 1 | Corresponding interrupt of IRR is masked (Initial value) |

### (6) Transmit Error Counter (TEC) and Receive Error Counter (REC)

The Transmit Error Counter (TEC) and Receive Error Counter (REC) is a 16-bit read/(write) register that functions as a counter indicating the number of transmit/receive message errors on the CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [1], [2], [3] and [4]. When not in (Write Error Counter) test mode this register is read only, and can only be modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) or entering to bus off.

In Write Error Counter test mode (i.e. TST[2:0] = 3'b100), it is possible to write to this register. The same value can only be written to TEC/REC, and the value written into TEC is set to TEC and REC. When writing to this register, RCAN-ET needs to be put into Halt Mode. This feature is only intended for test purposes.

- TEC/REC (Address = H'00C)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TEC7 | TEC6 | TEC5 | TEC4 | TEC3 | TEC2 | TEC1 | TEC0 | REC7 | REC6 | REC5 | REC4 | REC3 | REC2 | REC1 | REC0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |

Note: * It is only possible to write the value in test mode when TST[2:0] in MCR is 3'b100.
REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence.

### 19.3.4 RCAN-ET Mailbox Registers

The following sections describe RCAN-ET Mailbox registers that control / flag individual Mailboxes. The address is mapped as follows.

**Important:** LongWord access is carried out as two consecutive Word accesses.

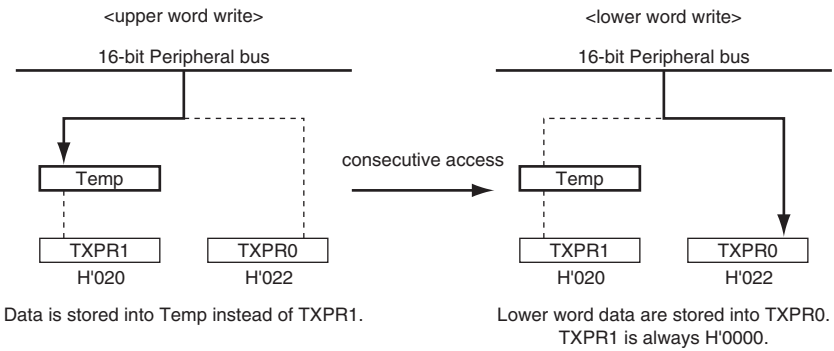
| Description | Address | Name | Access Size (bits) |
|---|---|---|---|
| Transmit Pending 1 | H'020 | TXPR1 | LW |
| Transmit Pending 0 | H'022 | TXPR0 | — |
| | H'024 | | |
| | H'026 | | |
| | H'028 | | |
| Transmit Cancel 0 | H'02A | TXCR0 | |
| | H'02C | | |
| | H'02E | | |
| | H'030 | | |
| Transmit Acknowledge 0 | H'032 | TXACK0 | Word |
| | H'034 | | |
| | H'036 | | |
| | H'038 | | |
| Abort Acknowledge 0 | H'03A | ABACK0 | Word |
| | H'03C | | |
| | H'03E | | |
| | H'040 | | |
| Data Frame Receive Pending 0 | H'042 | RXPR0 | Word |
| | H'044 | | |
| | H'046 | | |
| | H'048 | | |
| Remote Frame Receive Pending 0 | H'04A | RFPR0 | Word |
| | H'04C | | |
| | H'04E | | |
| | H'050 | | |
| Mailbox Interrupt Mask Register 0 | H'052 | MBIMR0 | Word |
| | H'054 | | |
| | H'056 | | |
| | H'058 | | |
| Unread message Status Register 0 | H'05A | UMSR0 | Word |
| | H'05C | | |
| | H'05E | | |

**Figure 19.7 RCAN-ET Mailbox Registers**

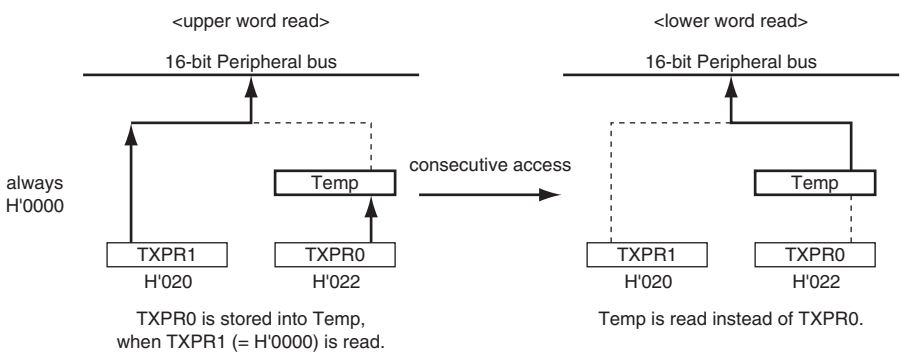
### (1) Transmit Pending Register (TXPR1, TXPR0)

The concatenation of TXPR1 and TXPR0 is a 32-bit register that contains any transmit pending flags for the CAN module. In the case of 16-bit bus interface, Long Word access is carried out as two consecutive word accesses.

<Longword Write Operation>




<Longword Read Operation>




The TXPR1 register cannot be modified and it is always fixed to '0'. The TXPR0 controls Mailbox-15 to Mailbox-1. The CPU may set the TXPR bits to affect any message being considered for transmission by writing a '1' to the corresponding bit location. Writing a '0' has no effect, and TXPR cannot be cleared by writing a '0' and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the CPU to determine which, if any, transmissions are pending or in progress. In effect there is a transmit pending bit for all Mailboxes except for the Mailbox-0. Writing a '1' to a bit location when the mailbox is not configured to transmit is not allowed.

The RCAN-ET will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. The TXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and RCAN-ET automatically tries to transmit it again unless its DART bit (Disable Automatic Re-Transmission) is set in the Message-Control of the corresponding Mailbox. In such case (DART set), the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR8) and the correspondent bit within the Abort Acknowledgement Register (ABACK).

If the status of the TXPR changes, the RCAN-ET shall ensure that in the identifier priority scheme (MCR2=0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to section 19.4, Application Note.

When the RCAN-ET changes the state of any TXPR bit position to a '0', an empty slot interrupt (IRR8) may be generated. This indicates that either a successful or an aborted mailbox transmission has just been made. If a message transmission is successful it is signalled in the TXACK register, and if a message transmission abortion is successful it is signalled in the ABACK register. By checking these registers, the contents of the Message of the corresponding Mailbox may be modified to prepare for the next transmission.

- TXPR1

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TXPR1[15:0] | | | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |

Note: * Any write operation is ignored.

Read value is always H'0000. Long word access is mandatory when reading or writing TXPR1/TXPR0. Writing any value to TXPR1 is allowed, however, write operation to TXPR1 has no effect.

- TXPR0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TXPR0[15:1] | | | | | | | | | | | | | | | 0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | - |

Note: * it is possible only to write a '1' for a Mailbox configured as transmitter.

**Bits 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)<br>[Clearing Condition] Completion of message transmission or message transmission abortion (automatically cleared) |
| 1 | Transmission request made for corresponding mailbox |

**Bit 0 — Reserved**: This bit is always '0' as this is a receive-only Mailbox. Writing a '1' to this bit position has no effect. The returned value is '0'.

### (2) Transmit Cancel Register (TXCR0)

TXCR0 is a 16-bit read / conditionally-write registers. The TXCR0 controls Mailbox-15 to Mailbox-1.This register is used by the CPU to request the pending transmission requests in the TXPR to be cancelled. To clear the corresponding bit in the TXPR the CPU must write a '1' to the bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXCR bits, and sets the corresponding ABACK bit. However, once a Mailbox has started a transmission, it cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABACK bit. If an attempt is made by the CPU to clear a mailbox transmission that is not transmit-pending it has no effect. In this case the CPU will be not able at all to set the TXCR flag.

- TXCR0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TXCR0[15:1] | | | | | | | | | | | | | | | 0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | - |

Note: * Only writing a '1' to a Mailbox that is requested for transmission and is configured as transmit.

**Bits 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)<br>[Clearing Condition] Completion of transmit message cancellation (automatically cleared) |
| 1 | Transmission cancellation request made for corresponding mailbox |

**Bit 0**: This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

### (3) Transmit Acknowledge Register (TXACK0)

The TXACK0 is a 16-bit read / conditionally-write registers. This register is used to signal to the CPU that a mailbox transmission has been successfully made. When a transmission has succeeded the RCAN-ET sets the corresponding bit in the TXACK register. The CPU may clear a TXACK bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect.

- TXACK0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TXACK0[15:1] | | | | | | | | | | | | | | | 0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | - |

Note: * Only when writing a '1' to clear.

**Bits 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)<br>[Setting Condition] Completion of message transmission for corresponding mailbox |

**Bit 0**: This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

### (4) Abort Acknowledge Register (ABACK0)

The ABACK0 is a 16-bit read / conditionally-write registers. This register is used to signal to the CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded the RCAN-ET sets the corresponding bit in the ABACK register. The CPU may clear the Abort Acknowledge bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect. An ABACK bit position is set by the RCAN-ET to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ABACK0[15:1] | | | | | | | | | | | | | | | 0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | - |

Note: * Only when writing a '1' to clear.

**Bits 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)<br>[Setting Condition] Completion of transmission cancellation for corresponding mailbox |

**Bit 0**: This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

### (5) Data Frame Receive Pending Register (RXPR0)

The RXPR0 is a 16-bit read / conditionally-write registers. The RXPR is a register that contains the received Data Frames pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is set in the RXPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data Frame Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by receiving Data Frames and not by receiving Remote frames.

- RXPR0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RXPR0[15:0] | | | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |

Note: * Only when writing a '1' to clear.

**Bits 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<br>[Setting Condition] Completion of Data Frame receive on corresponding mailbox |

### (6) Remote Frame Receive Pending Register (RFPR0)

The RFPR0 is a 16-bit read / conditionally-write registers. The RFPR is a register that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is set in the RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. In effect there is a bit position for all mailboxes. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a RFPR bit is set, it also sets IRR2 (Remote Frame Request Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Please note that these bits are only set by receiving Remote Frames and not by receiving Data frames.

- RFPR0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFPR0[15:0] | | | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |

Note: * Only when writing a '1' to clear.

**Bits 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<br>[Setting Condition] Completion of remote frame receive in corresponding mailbox |

### (7) Mailbox Interrupt Mask Register (MBIMR)

The MBIMR1 and MBIMR0 are 16-bit read / write registers. The MBIMR only prevents the setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received Interrupt, IRR[2] – Remote Frame Request Interrupt, IRR[8] – Mailbox Empty Interrupt, and IRR[9] – Message OverRun/OverWrite Interrupt. If a mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR[2] and IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or UMSR. Similarly when a mailbox has been configured for transmission, a mask prevents the generation of an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or abortion of transmission (IRR[8]), however, it does not prevent the RCAN-ET from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and it does not prevent the RCAN-ET from clearing the corresponding TXPR/TXCR bit + setting the ABACK bit for abortion of the transmission.

A mask is set by writing a '1' to the corresponding bit position for the mailbox activity to be masked. At reset all mailbox interrupts are masked.

- MBIMR0

| Bit: | 15 | 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 |

**Bits 15 to 0:** Enable or disable interrupt requests from individual Mailbox-15 to Mailbox-0 respectively.

| Bit[15:0]: MBIMR0 | Description |
|---|---|
| 0 | Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled |
| 1 | Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value) |

### (8) Unread Message Status Register (UMSR)

This register is a 16-bit read/conditionally write register and it records the mailboxes whose contents have not been accessed by the CPU prior to a new message being received. If the CPU has not cleared the corresponding bit in the RXPR or RFPR when a new message for that mailbox is received, the corresponding UMSR bit is set to '1'. This bit may be cleared by writing a '1' to the corresponding bit location in the UMSR. Writing a '0' has no effect.

If a mailbox is configured as transmit box, the corresponding UMSR will not be set.

- UMSR0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | UMSR0[15:0] | | | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |

**Bits 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<br>[Setting Condition] When a new message is received before RXPR or RFPR is cleared |

## 19.4 Application Note

### 19.4.1 Test Mode Settings

The RCAN-ET has various test modes. The register TST[2:0] (MCR[10:8]) is used to select the RCAN-ET test mode. The default (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.

| Bit10: TST2 | Bit9: TST1 | Bit8: TST0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | Normal Mode (initial value) |
| 0 | 0 | 1 | Listen-Only Mode (Receive-Only Mode) |
| 0 | 1 | 0 | Self Test Mode 1 (External) |
| 0 | 1 | 1 | Self Test Mode 2 (Internal) |
| 1 | 0 | 0 | Write Error Counter |
| 1 | 0 | 1 | Error Passive Mode |
| 1 | 1 | 0 | setting prohibited |
| 1 | 1 | 1 | setting prohibited |

Normal Mode: RCAN-ET operates in the normal mode.

Listen-Only Mode: ISO-11898 requires this mode for baud rate detection. The Error Counters are cleared and disabled so that the TEC/REC does not increase the values, and the Tx Output is disabled so that RCAN-ET does not generate error frames or acknowledgment bits. IRR13 is set when a message error occurs.

Self Test Mode 1: RCAN-ET generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The Rx/Tx pins must be connected to the CAN bus.

Self Test Mode 2: RCAN-ET generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The Rx/Tx pins do not need to be connected to the CAN bus or any external devices, as the internal Tx is looped back to the internal Rx. Tx pin outputs only recessive bits and Rx pin is disabled.

Write Error Counter: TEC/REC can be written in this mode. RCAN-ET can be forced to become an Error Passive mode by writing a value greater than 127 into the Error Counters. The value written into TEC is used to write into REC, so only the same value can be set to these registers. Similarly, RCAN-ET can be forced to become an Error Warning by writing a value greater than 95 into them.

RCAN-ET needs to be in Halt Mode when writing into TEC/REC (MCR1 must be "1" when writing to the Error Counter). Furthermore this test mode needs to be exited prior to leaving Halt mode.Error Passive Mode: RCAN-ET can be forced to enter Error Passive mode.

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

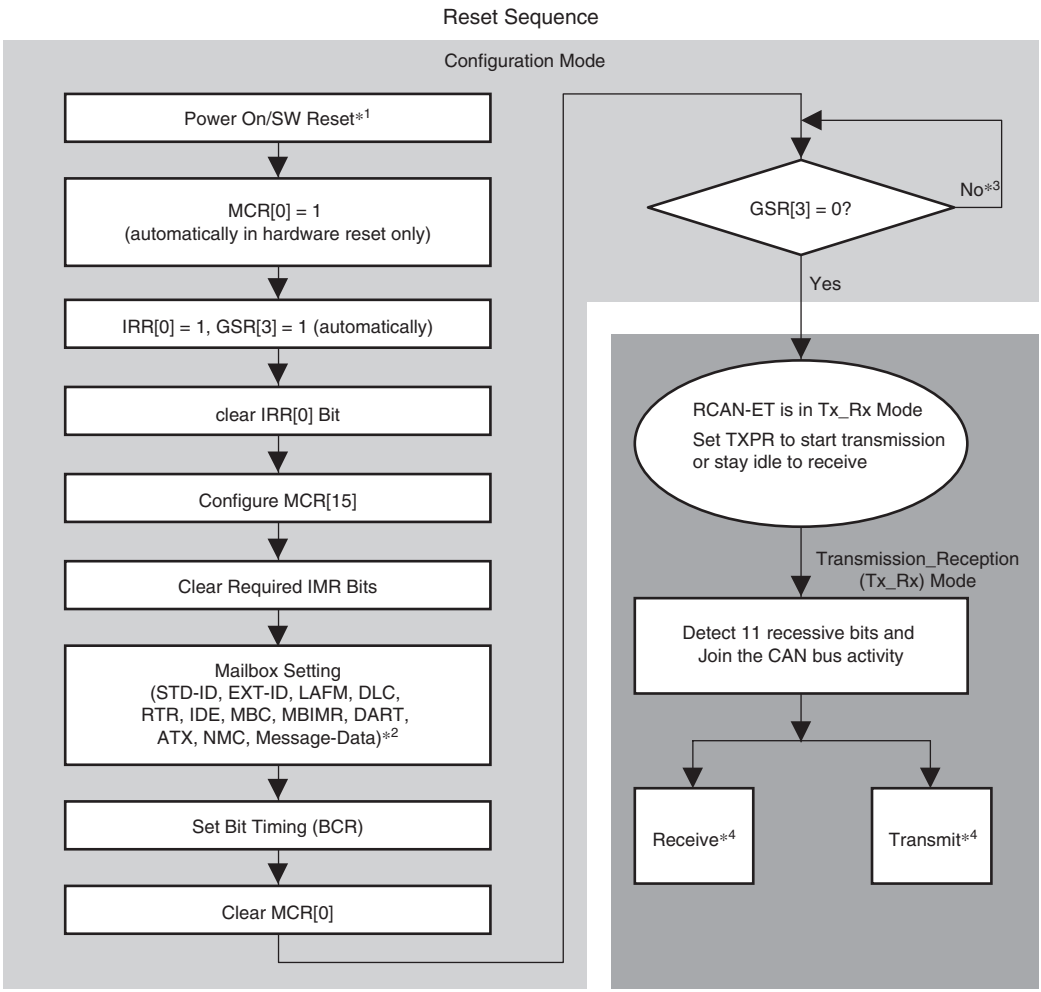
When message error occurs, IRR13 is set in all test modes.

### 19.4.2 Configuration of RCAN-ET

RCAN-ET is considered in configuration mode or after a H/W (Power On Reset)/ S/W (MCR[0]) reset or when in Halt mode. In both conditions RCAN-ET cannot join the CAN Bus activity and configuration changes have no impact on the traffic on the CAN Bus.

- After a Reset request

  The following sequence must be implemented to configure the RCAN-ET after (S/W or H/W) reset. After reset, all the registers are 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 initialise all the mailboxes enabled by MBC.
3. It takes about one bit time quantum for the value to reach 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 19.8 Reset Sequence**

- Halt mode

  When RCAN-ET is in Halt mode, it cannot take part to the CAN bus activity. Consequently the user can modify all the requested registers without influencing existing traffic on the CAN Bus. It is important for this that the user waits for the RCAN-ET to be in halt mode before to modify the requested registers - note that the transition to Halt Mode is not always immediate (transition will occurs when the CAN Bus is idle or in intermission). After RCAN-ET transit to Halt Mode, GSR4 is set.

  Once the configuration is completed the Halt request needs to be released. RCAN-ET will join CAN Bus activity after the detection of 11 recessive bits on the CAN Bus.

- Sleep mode

  When RCAN-ET is in sleep mode the clock for the main blocks of the IP is stopped in order to reduce power consumption. Only the following user registers are clocked and can be accessed: MCR, GSR, IRR and IMR. Interrupt related to transmission (TXACK and ABACK) and reception (RXPR and RFPR) cannot be cleared when in sleep mode (as TXACK, ABACK, RXPR and RFPR are not accessible) and must to be cleared beforehand.

  The following diagram shows the flow to follow to move RCAN-ET into sleep mode.

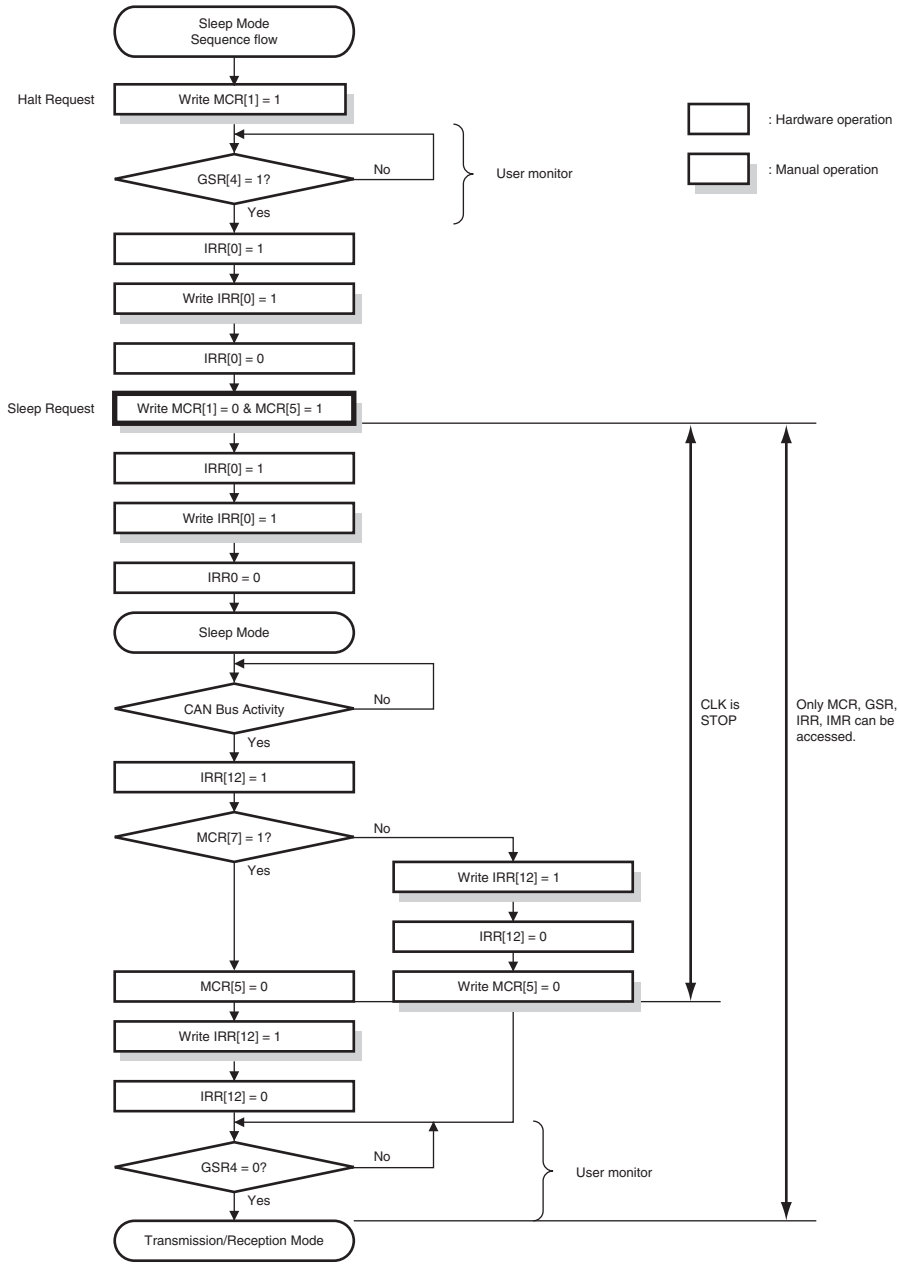

Sleep Mode
Sequence flow
Halt Request
Write MCR[1] = 1
: Hardware operation
: Manual operation
GSR[4] = 1?
No
User monitor
Yes
IRR[0] = 1
Write IRR[0] = 1
IRR[0] = 0
Sleep Request
Write MCR[1] = 0 & MCR[5] = 1
IRR[0] = 1
Write IRR[0] = 1
IRR0 = 0
Sleep Mode
CAN Bus Activity
No
Yes
CLK is
STOP
Only MCR, GSR,
IRR, IMR can be
accessed.
IRR[12] = 1
MCR[7] = 1?
No
Yes
Write IRR[12] = 1
IRR[12] = 0
MCR[5] = 0
Write MCR[5] = 0
Write IRR[12] = 1
IRR[12] = 0
GSR4 = 0?
No
User monitor
Yes
Transmission/Reception Mode

Figure 19.9 - Halt Mode / Sleep Mode shows allowed state transition.

— Please don't set MCR5 (Sleep Mode) without entering Halt Mode.

— After MCR1 is set, please don't clear it before GSR4 is set and RCAN-ET enters Halt Mode.

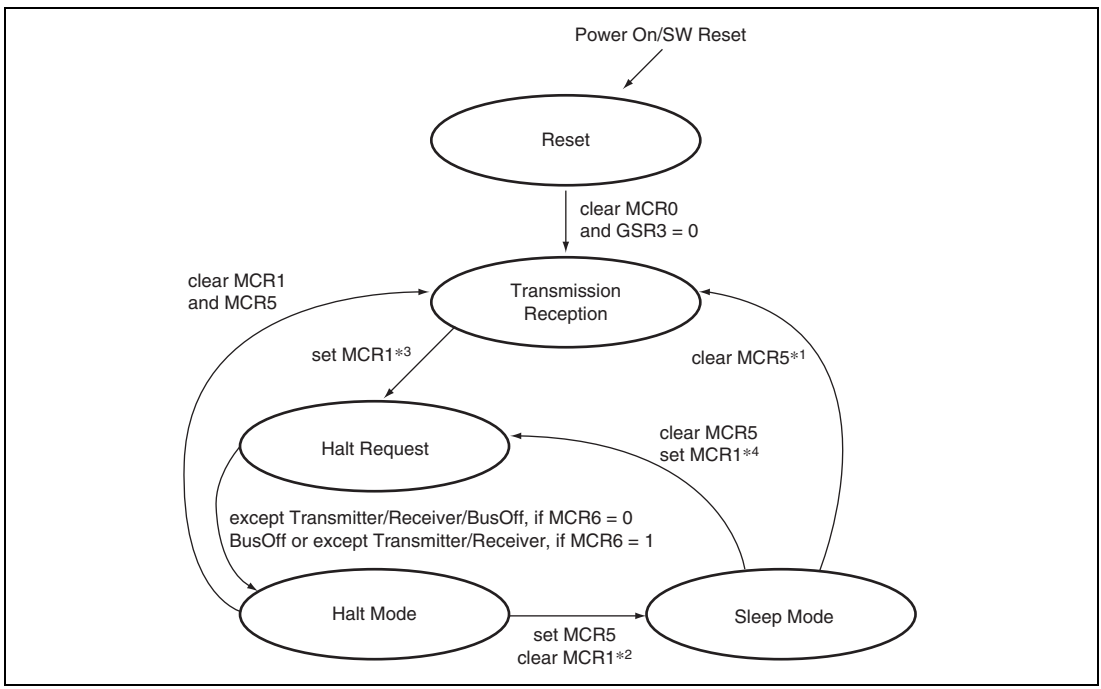



**Figure 19.9 Halt Mode / Sleep Mode**

Notes: 1. MCR5 can be cleared by automatically by detecting a dominant bit on the CAN Bus if MCR7 is set or by writing "0"

2. MCR1 is cleared in SW. Clearing MCR1 and setting MCR5 have to be carried out by the same instruction.

3. MCR1 must not be cleared in SW, before GSR4 is set. MCR1 can be set automatically in HW when RCAN-ET moves to Bus Off and MCR14 and MCR6 are both set.

4. When MCR5 is cleared and MCR1 is set at the same time, RCAN-ET moves to Halt Request. Right after that, it moves to Halt Mode with no reception/transmission.

The following table shows conditions to access registers.

| | RCAN-ET Registers | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Status Mode** | **MCR GSR** | **IRR IMR** | **BCR** | **MBIMR** | **Flag_register** | **mailbox (ctrl0, LAFM)** | | **mailbox (data)** | **mailbox (ctrl1)** | |
| Reset | yes | yes | yes | yes | yes | yes | | yes | yes | |
| Transmission<br>Reception<br>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.

### 19.4.3 Message Transmission Sequence

- Message Transmission Request

  The following sequence is an example to transmit a CAN frame onto the bus. As described in the previous register section, please note that IRR8 is set when one of the TXACK or ABACK bits is set, meaning one of the Mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, the GSR2 means that there is currently no transmission request made (No TXPR flags set).

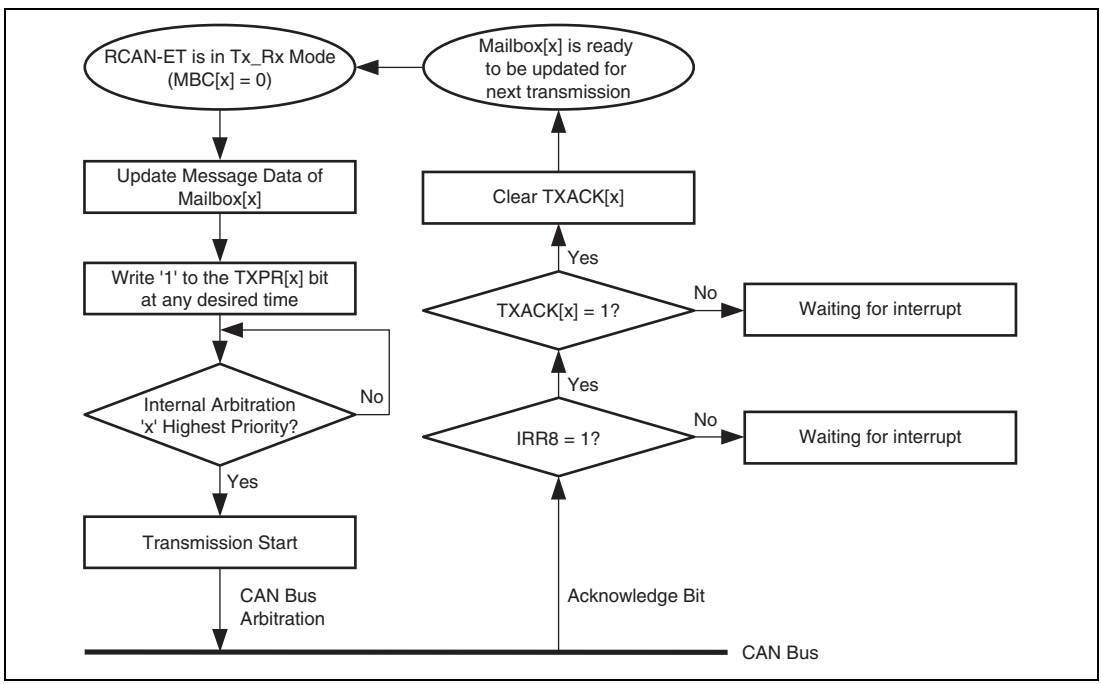



**Figure 19.10 Transmission Request**

- Internal Arbitration for transmission

  The following diagram explains how RCAN-ET manages to schedule transmission-requested messages in the correct order based on the CAN identifier. 'Internal arbitration' picks up the highest priority message amongst transmit-requested messages.

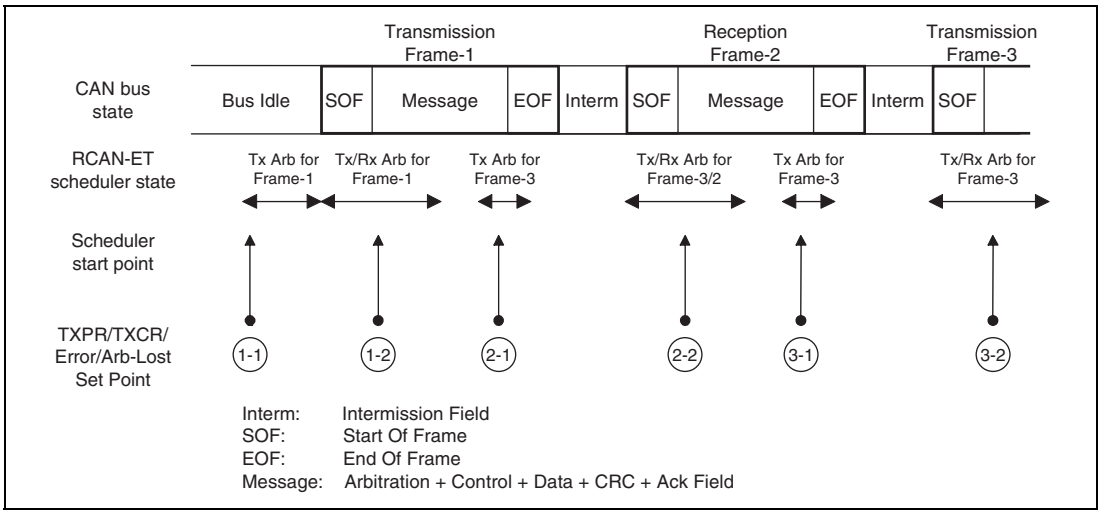



**Figure 19.11 Internal Arbitration for Transmission**

The RCAN-ET has two state machines. One is for transmission, and the other is for reception.

1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.

1-2: Operations for both transmission and reception starts at SOF. Since there is no reception frame, RCAN-ET becomes transmitter.

2-1: At crc delimiter, internal arbitration to search next message transmitted starts.

2-2: Operations for both transmission and reception starts at SOF. Because of a reception frame with higher priority, RCAN-ET becomes receiver. Therefore, Reception is carried out instead of transmitting Frame-3.

3-1: At crc delimiter, internal arbitration to search next message transmitted starts.

3-2: Operations for both transmission and reception starts at SOF. Since a transmission frame has higher priority than reception one, RCAN-ET becomes transmitter.

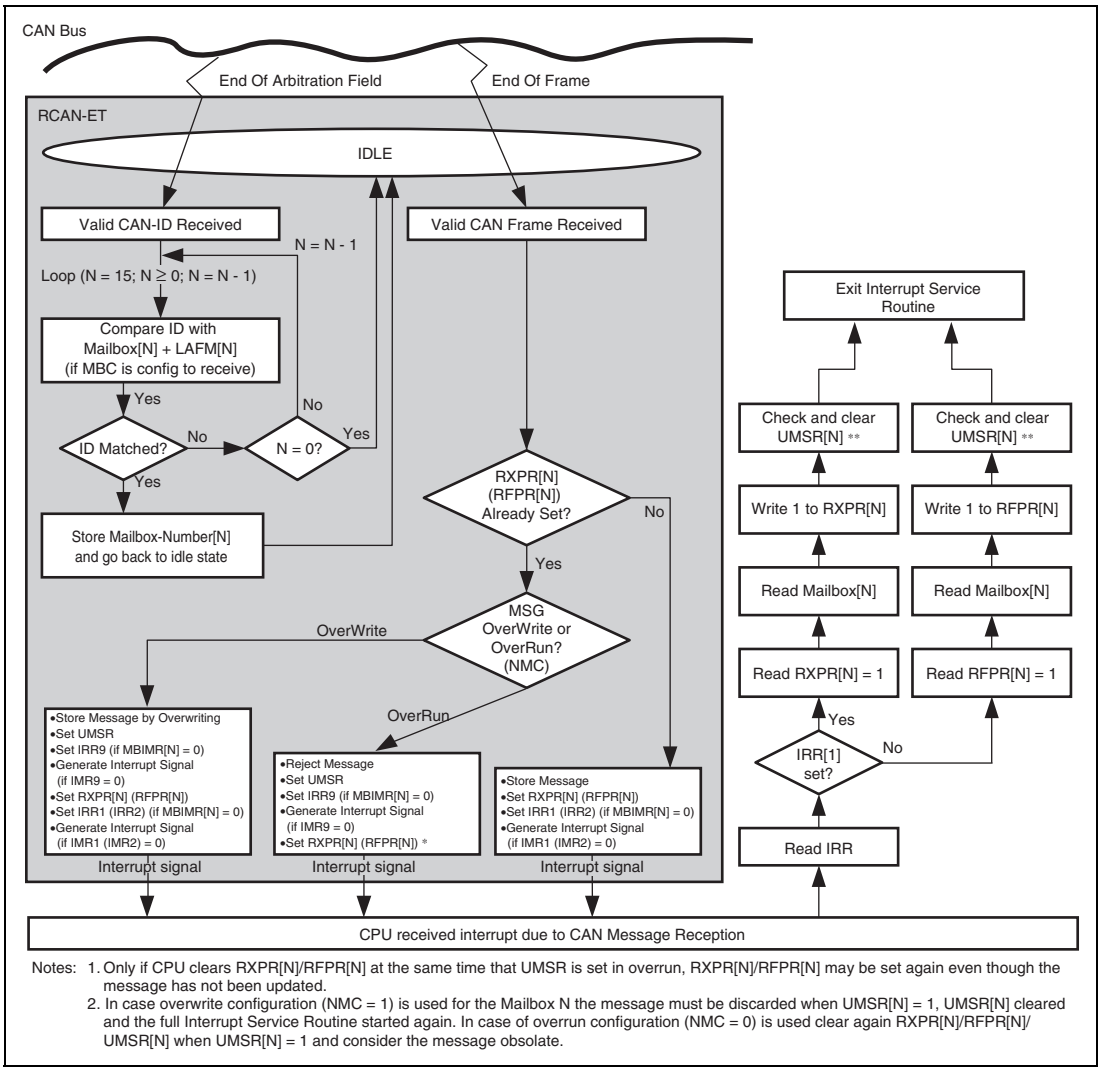
Internal arbitration for the next transmission is also performed at the beginning of each error delimiter in case of an error is detected on the CAN Bus. It is also performed at the beginning of error delimiters following overload frame.

As the arbitration for transmission is performed at CRC delimiter, in case a remote frame request is received into a Mailbox with ATX=1 the answer can join the arbitration for transmission only at the following Bus Idle, CRC delimiter or Error Delimiter.

Depending on the status of the CAN bus, following the assertion of the TXCR, the corresponding Message abortion can be handled with a delay of maximum 1 CAN Frame.

### 19.4.4 Message Receive Sequence

The diagram below shows the message receive sequence.




Notes: 1. Only if CPU clears RXPR[N]/RFPR[N] at the same time that UMSR is set in overrun, RXPR[N]/RFPR[N] may be set again even though the message has not been updated.
2. In case overwrite configuration (NMC = 1) is used for the Mailbox N the message must be discarded when UMSR[N] = 1, UMSR[N] cleared and the full Interrupt Service Routine started again. In case of overrun configuration (NMC = 0) is used clear again RXPR[N]/RFPR[N]/UMSR[N] when UMSR[N] = 1 and consider the message obsolete.

**Figure 19.12 Message Receive Sequence**

When RCAN-ET recognises the end of the Arbitration field while receiving a message, it starts comparing the received identifier to the identifiers set in the Mailboxes, starting from Mailbox-15 down to Mailbox-0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox-15 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-14 (if configured as receive). Once RCAN-ET finds a matching identifier, it stores the number of Mailbox-[N] into an internal buffer, stops the search, and goes back to idle state, waiting for the EndOfFrame (EOF) to come. When the $6^{th}$ bit of EOF is notified by the CAN Interface logic, the received message is written or abandoned, depending on the NMC bit. No modification of configuration during communication is allowed. Entering Halt Mode is one of ways to modify configuration. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of a received message matches to ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

With regards to the reception of data and remote frames described in the above flow diagram the clearing of the UMSR flag after the reading of IRR is to detect situations where a message is overwritten by a new incoming message stored in the same mailbox while the interrupt service routine is running. If during the final check of UMSR a overwrite condition is detected the message needs to be discarded and read again.

In case UMSR is set and the Mailbox is configured for overrun (NMC = 0) the message is still valid, however it is obsolete as it is not reflecting the latest message monitored on the CAN Bus. Please access the full Mailbox content before clearing the related RXPR/RFPR flag.

Please note that in the case a received remote frame is overwritten by a data frame, both the remote frame request interrupt (IRR2) and data frame received interrupt (IRR1) and also the Receive Flags (RXPR and RFPR) are set. In an analogous way, the overwriting of a data frame by a remote frame, leads to setting both IRR2 and IRR1.

In the Overrun Mode (NMC = '0'), only the first Mailbox will cause the flags to be asserted. So, if a Data Frame is initially received, then RXPR and IRR1 are both asserted. If a Remote Frame is then received before the Data Frame has been read, then RFPR and IRR2 are NOT set. In this case UMSR of the corresponding Mailbox will still be set.

### 19.4.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

- Change configuration of transmit box

  Two cases are possible.

  — Change of ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART

  This change is possible only when MBC=3'b000. Confirm that the corresponding TXPR is not set. The configuration (except MBC bit) can be changed at any time.

  — Change from transmit to receive configuration (MBC)

  Confirm that the corresponding TXPR is not set. The configuration can be changed only in Halt or reset state. Please note that it might take longer for RCAN-ET to transit to halt state if it is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-ET will not be able to receive/transmit messages during the Halt state.

  In case RCAN-ET is in the Bus Off state the transition to halt state depends on the configuration of the bit 6 of MCR and also bit and 14 of MCR.

- Change configuration (ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART, MBC) of receiver box or Change receiver box to transmitter box

  The configuration can be changed only in Halt Mode.

  RCAN-ET will not lose a message if the message is currently on the CAN bus and RCAN-ET is a receiver. RCAN-ET will be moving into Halt Mode after completing the current reception. Please note that it might take longer if RCAN-ET is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-ET will not be able to receive/transmit messages during the Halt Mode.

  In case RCAN-ET is in the Bus Off state the transition to halt mode depends on the configuration of the bit 6 and 14 of MCR.

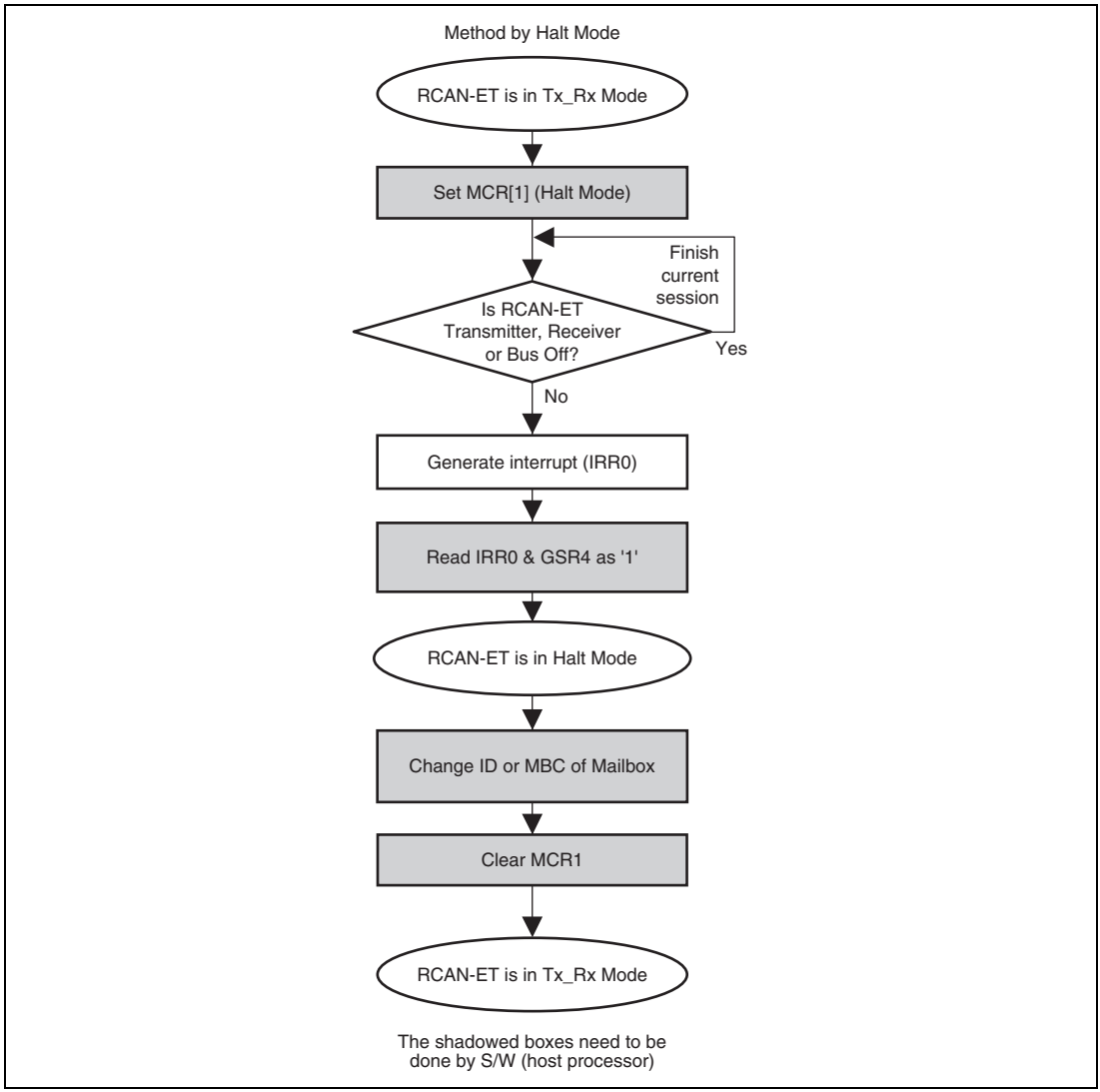



**Figure 19.13 Change ID of Receive Box or Change Receive Box to Transmit Box**

## 19.5 Interrupt Sources

Table 19.2 lists the RCAN-ET interrupt sources. With the exception of the reset processing interrupt (IRR0) by a power-on reset, these sources can be masked. Masking is implemented using the mailbox interrupt mask register 0 (MBIMR0) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 6, Interrupt Controller (INTC).

**Table 19.2 RCAN-ET Interrupt Sources**

| Module | Interrupt | Description | Interrupt Flag | DTC Activation |
|---|---|---|---|---|
| RCAN-ET_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 | |
| | RM0_0*[2] | Data frame reception | IRR1*[3] | Possible*[4] |
| | RM1_0*[2] | Remote frame reception | IRR2*[3] | |
| | SLE_0 | Message transmission/transmission disabled (slot empty) | IRR8 | Not possible |

Notes: 1. Available only in Test Mode.
2. RM0_0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1_0 is an interrupt generated by the remote request pending flag for mailbox n (RFPR0[n]) or the data frame receive flag for mailbox n (RXPR0[n]) (n = 1 to 15).
3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 15, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 15.
4. The DTC can be activated only by the RM0_0 interrupt.

## 19.6 DTC Interface

The DTC can be activated by the reception of a message in RCAN-ET mailbox 0. When DTC transfer ends after DTC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-ET cannot be sent to the CPU in this case. Figure 19.14 shows a DTC transfer flowchart.

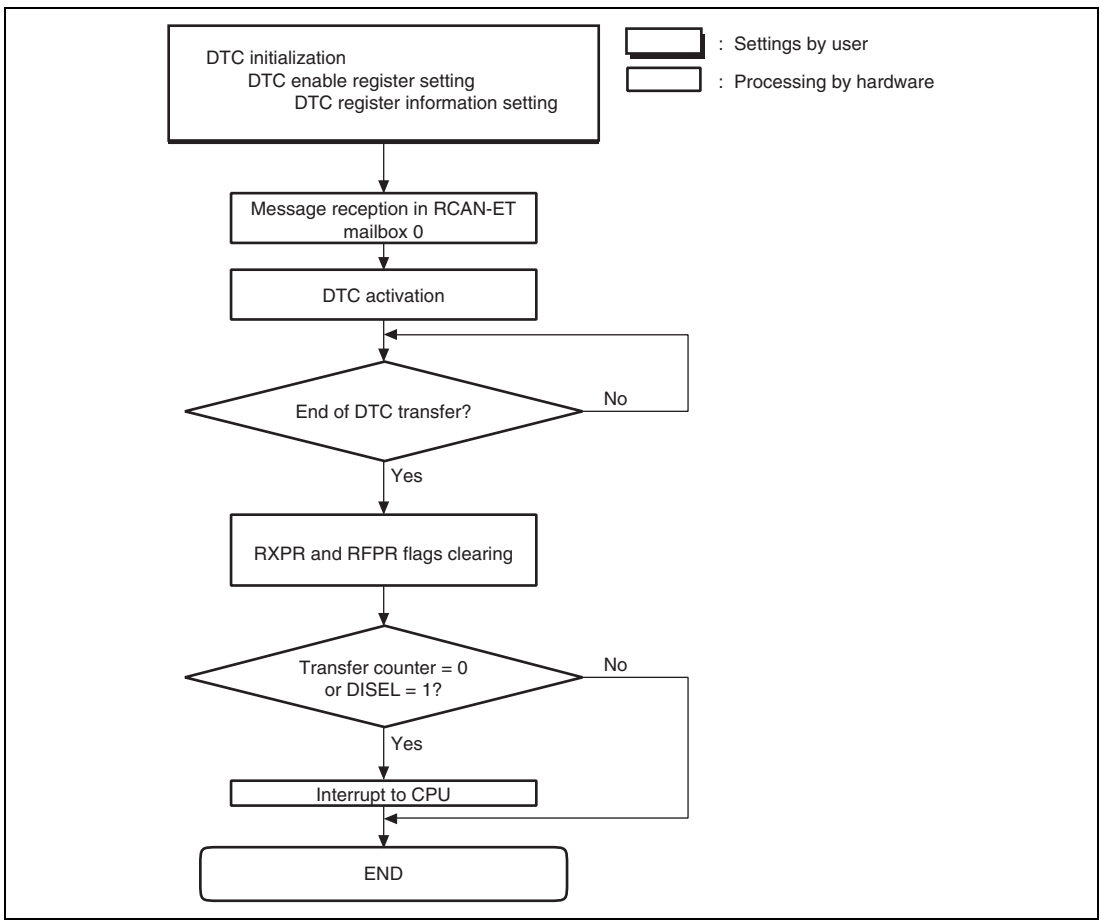



**Figure 19.14 DTC Transfer Flowchart**

## 19.7 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. Figure 19.15 shows a sample connection diagram.

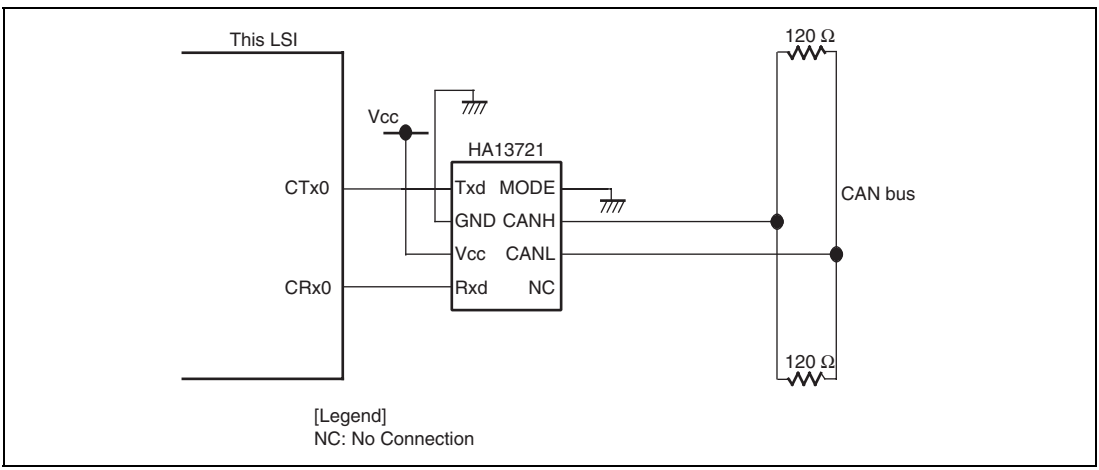



**Figure 19.15 High-Speed CAN Interface Using HA13721**

## 19.8 Usage Notes

### 19.8.1 Module Stop Mode

The clock supply to RCAN-ET can be stopped or started by using the standby control register 3 (STBCR3). With the initial value, the clock supply is stopped. Access to the RCAN-ET registers should be made only after releasing RCAN-ET from module stop mode.

### 19.8.2 Reset

RCAN-ET can be reset by hardware reset or software reset.

- Hardware reset
  RCAN-ET is reset to the initial state by power-on reset or on entering module stop mode or software standby mode.
- Software reset
  By setting the MCR0 bit in Master Control Register (MCR), RCAN-ET registers, excluding the MCR0 bit, and the CAN communication circuitry are initialized.

Since the IRR0 bit in Interrupt Request Register (IRR) is set by the initialization upon reset, it should be cleared while RCAN-ET is in configuration mode during the reset sequence.

The areas except for message control field 1 (CONTROL1) of mailboxes are not initialized by reset because they are in RAM. After power-on reset, all mailboxes should be initialized while RCAN-ET is in configuration mode during the reset sequence.

### 19.8.3 CAN Sleep Mode

In CAN sleep mode, the clock supply to the major parts in the module is stopped. Therefore, do not make access in CAN sleep mode except for access to the MCR, GSR, IRR, and IMR registers.

### 19.8.4 Register Access

If the mailbox area is accessed while the CAN communication circuitry in RCAN-ET is storing a received CAN bus frame in a mailbox, a 0 to five peripheral clock cycles of wait state is generated.

### 19.8.5 Interrupts

As shown in table 19.2, a Mailbox 0 receive interrupt can activate the DTC. If configured such that the DTC is activated by a Mailbox 0 receive interrupt and clearing of the interrupt source flag upon DTC transfer is enabled, use block transfer mode and read the whole Mailbox 0 message up to the message control field 1 (CONTROL1).

# Section 20 Pin Function Controller (PFC)

The pin function controller (PFC) is composed of registers that are used to select the functions of multiplexed pins and assign pins to be inputs or outputs. Tables 20.1 to 20.9 list the multiplexed pins of this LSI.

Tables 20.10 to 20.12 list the pin functions in each operating mode.

**Table 20.1 SH7131/SH7136 Multiplexed Pins (Port A)**

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) |
|---|---|---|---|---|
| A | PA0 I/O (port) | $\overline{\text{POE0}}$ input (POE) | RXD0 input (SCI) | — |
| | PA1 I/O (port) | $\overline{\text{POE1}}$ input (POE) | TXD0 output (SCI) | — |
| | PA2 I/O (port) | IRQ0 input (INTC) | $\overline{\text{POE2}}$ input (POE) | SCK0 I/O (SCI) |
| | PA3 I/O (port) | IRQ1 input (INTC) | RXD1 input (SCI) | — |
| | PA4 I/O (port) | IRQ2 input (INTC) | TXD1 output (SCI) | — |
| | PA5 I/O (port) | IRQ3 input (INTC) | SCK1 I/O (SCI) | — |
| | PA6 I/O (port) | $\overline{\text{UBCTRG}}$ output (UBC)* | TCLKA input (MTU2) | $\overline{\text{POE4}}$ input (POE) |
| | PA7 I/O (port) | TCLKB input (MTU2) | $\overline{\text{POE5}}$ input (POE) | SCK2 I/O (SCI) |
| | PA8 I/O (port) | TCLKC input (MTU2) | $\overline{\text{POE6}}$ input (POE) | RXD2 input (SCI) |
| | PA9 I/O (port) | TCLKD input (MTU2) | $\overline{\text{POE8}}$ input (POE) | TXD2 output (SCI) |
| | PA10 I/O (port) | RXD0 input (SCI) | — | — |
| | PA11 I/O (port) | TXD0 output (SCI) | $\overline{\text{ADTRG}}$ input (A/D) | — |
| | PA12 I/O (port) | SCK0 I/O (SCI) | $\overline{\text{SCS}}$ I/O (SSU) | — |
| | PA13 I/O (port) | SCK1 I/O (SCI) | SSCK I/O (SSU) | — |
| | PA14 I/O (port) | RXD1 input (SCI) | SSI I/O (SSU) | — |
| | PA15 I/O (port) | TXD1 output (SCI) | SSO I/O (SSU) | — |

Note: * Function enabled on the SH7136 only.

## Table 20.2 SH7132/SH7137 Multiplexed Pins (Port A)

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) | Function 5 (Related Module) |
|---|---|---|---|---|---|
| A | PA0 I/O (port) | A0 output (BSC) | $\overline{\text{POE0}}$ input (POE) | RXD0 input (SCI) | — |
| | PA1 I/O (port) | A1 output (BSC) | $\overline{\text{POE1}}$ input (POE) | TXD0 output (SCI) | — |
| | PA2 I/O (port) | A2 output (BSC) | IRQ0 input (INTC) | $\overline{\text{POE2}}$ input (POE) | SCK0 I/O (SCI) |
| | PA3 I/O (port) | A3 output (BSC) | IRQ1 input (INTC) | RXD1 input (SCI) | — |
| | PA4 I/O (port) | A4 output (BSC) | IRQ2 input (INTC) | TXD1 output (SCI) | — |
| | PA5 I/O (port) | A5 output (BSC) | IRQ3 input (INTC) | SCK1 I/O (SCI) | — |
| | PA6 I/O (port) | $\overline{\text{RD}}$ output (BSC) | $\overline{\text{UBCTRG}}$ output (UBC)* | TCLKA input (MTU2) | $\overline{\text{POE4}}$ input (POE) |
| | PA7 I/O (port) | TCLKB input (MTU2) | $\overline{\text{POE5}}$ input (POE) | SCK2 I/O (SCI) | — |
| | PA8 I/O (port) | $\overline{\text{WRL}}$ output (BSC) | TCLKC input (MTU2) | $\overline{\text{POE6}}$ input (POE) | RXD2 input (SCI) |
| | PA9 I/O (port) | $\overline{\text{WAIT}}$ input (BSC) | TCLKD input (MTU2) | $\overline{\text{POE8}}$ input (POE) | TXD2 output (SCI) |
| | PA10 I/O (port) | A6 output (BSC) | RXD0 input (SCI) | — | — |
| | PA11 I/O (port) | A7 output (BSC) | TXD0 output (SCI) | $\overline{\text{ADTRG}}$ input (A/D) | — |
| | PA12 I/O (port) | A8 output (BSC) | SCK0 I/O (SCI) | $\overline{\text{SCS}}$ I/O (SSU) | — |
| | PA13 I/O (port) | A9 output (BSC) | SCK1 I/O (SCI) | SSCK I/O (SSU) | — |
| | PA14 I/O (port) | A10 output (BSC) | RXD1 input (SCI) | SSI I/O (SSU) | — |
| | PA15 I/O (port) | CK output (CPG) | TXD1 output (SCI) | SSO I/O (SSU) | — |

Note: * Function enabled on the SH7137 only.

## Table 20.3 SH7131/SH7136 Multiplexed Pins (Port B)

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) | Function 5 (Related Module) |
|---|---|---|---|---|---|
| B | PB2 I/O (port) | IRQ0 input (INTC) | $\overline{\text{POE0}}$ input (POE) | TIC5VS input (MTU2S) | SCL I/O ($I^2C2$) |
| | PB3 I/O (port) | IRQ1 input (INTC) | $\overline{\text{POE1}}$ input (POE) | TIC5V input (MTU2) | SDA I/O ($I^2C2$) |
| | PB4 I/O (port) | IRQ2 input (INTC) | $\overline{\text{POE4}}$ input (POE) | TIC5US input (MTU2S) | — |
| | PB5 I/O (port) | IRQ3 input (INTC) | $\overline{\text{POE5}}$ input (POE) | TIC5U input (MTU2) | — |
| | PB6 I/O (port) | CTx0 output (RCAN-ET) | — | — | — |
| | PB7 I/O (port) | CRx0 input (RCAN-ET) | — | — | — |

## Table 20.4 SH7132/SH7137 Multiplexed Pins (Port B)

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) | Function 5 (Related Module) | Function 6 (Related Module) |
|---|---|---|---|---|---|---|
| B | PB0 I/O (port) | $\overline{\text{BACK}}$ output (BSC) | TIC5WS input (MTU2S) | — | — | — |
| | PB1 I/O (port) | $\overline{\text{BREQ}}$ input (BSC) | TIC5W input (MTU2) | — | — | — |
| | PB2 I/O (port) | A16 output (BSC) | IRQ0 input (INTC) | $\overline{\text{POE0}}$ input (POE) | TIC5VS input (MTU2S) | SCL I/O ($I^2C2$) |
| | PB3 I/O (port) | A17 output (BSC) | IRQ1 input (INTC) | $\overline{\text{POE1}}$ input (POE) | TIC5V input (MTU2) | SDA I/O ($I^2C2$) |
| | PB4 I/O (port) | A18 output (BSC) | IRQ2 input (INTC) | $\overline{\text{POE4}}$ input (POE) | TIC5US input (MTU2S) | — |
| | PB5 I/O (port) | A19 output (BSC) | IRQ3 input (INTC) | $\overline{\text{POE5}}$ input (POE) | TIC5U input (MTU2) | — |
| | PB6 I/O (port) | $\overline{\text{WAIT}}$ input (BSC) | CTx0 output (RCAN-ET) | — | — | — |
| | PB7 I/O (port) | $\overline{\text{CS1}}$ output (BSC) | CRx0 input (RCAN-ET) | — | — | — |

## Table 20.5 SH7132/SH7137 Multiplexed Pins (Port D)

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) |
|---|---|---|---|---|
| D | PD0 I/O (port) | D0 I/O (BSC) | RXD0 input (SCI) | — |
| | PD1 I/O (port) | D1 I/O (BSC) | TXD0 output (SCI) | — |
| | PD2 I/O (port) | D2 I/O (BSC) | SCK0 I/O (SCI) | — |
| | PD3 I/O (port) | D3 I/O (BSC) | RXD1 input (SCI) | — |
| | PD4 I/O (port) | D4 I/O (BSC) | TXD1 output (SCI) | — |
| | PD5 I/O (port) | D5 I/O (BSC) | SCK1 I/O (SCI) | — |
| | PD6 I/O (port) | D6 I/O (BSC) | RXD2 input (SCI) | — |
| | PD7 I/O (port) | D7 I/O (BSC) | TXD2 output (SCI) | $\overline{\text{SCS}}$ I/O (SSU) |
| | PD8 I/O (port) | SCK2 I/O (SCI) | SSCK I/O (SSU) | — |
| | PD9 I/O (port) | SSI I/O (SSU) | — | — |
| | PD10 I/O (port) | SSO I/O (SSU) | — | — |

## Table 20.6 SH7131/SH7136 Multiplexed Pins (Port E)

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) |
|---|---|---|---|---|
| E | PE0 I/O (port) | TIOC0A I/O (MTU2) | — | — |
| | PE1 I/O (port) | TIOC0B I/O (MTU2) | RXD0 input (SCI) | — |
| | PE2 I/O (port) | TIOC0C I/O (MTU2) | TXD0 output (SCI) | — |
| | PE3 I/O (port) | TIOC0D I/O (MTU2) | SCK0 I/O (SCI) | — |
| | PE4 I/O (port) | TIOC1A I/O (MTU2) | RXD1 input (SCI) | — |
| | PE5 I/O (port) | TIOC1B I/O (MTU2) | TXD1 output (SCI) | — |
| | PE6 I/O (port) | TIOC2A I/O (MTU2) | SCK1 I/O (SCI) | — |
| | PE7 I/O (port) | TIOC2B I/O (MTU2) | — | — |
| | PE8 I/O (port) | TIOC3A I/O (MTU2) | — | — |
| | PE9 I/O (port) | TIOC3B I/O (MTU2) | — | — |
| | PE10 I/O (port) | TIOC3C I/O (MTU2) | — | — |
| | PE11 I/O (port) | TIOC3D I/O (MTU2) | — | — |
| | PE12 I/O (port) | TIOC4A I/O (MTU2) | — | — |
| | PE13 I/O (port) | TIOC4B I/O (MTU2) | $\overline{\text{MRES}}$ input (INTC) | — |
| | PE14 I/O (port) | TIOC4C I/O (MTU2) | — | — |
| | PE15 I/O (port) | TIOC4D I/O (MTU2) | $\overline{\text{IRQOUT}}$ output (INTC) | — |
| | PE16 I/O (port) | TIOC3BS I/O (MTU2S) | $\overline{\text{ASEBRKAK}}$ output (E10A)* | $\overline{\text{ASEBRK}}$ input (E10A)* |
| | PE17 I/O (port) | TIOC3DS I/O (MTU2S) | TCK input (H-UDI)* | — |
| | PE18 I/O (port) | TIOC4AS I/O (MTU2S) | TDI input (H-UDI)* | — |
| | PE19 I/O (port) | TIOC4BS I/O (MTU2S) | TDO output (H-UDI)* | — |
| | PE20 I/O (port) | TIOC4CS I/O (MTU2S) | TMS input (H-UDI)* | — |
| | PE21 I/O (port) | TIOC4DS I/O (MTU2S) | $\overline{\text{TRST}}$ input (H-UDI)* | — |

Note: * Function enabled on the SH7136 only.

## Table 20.7 SH7132/SH7137 Multiplexed Pins (Port E)

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) | Function 5 (Related Module) |
|---|---|---|---|---|---|
| E | PE0 I/O (port) | TIOC0A I/O (MTU2) | — | — | — |
| | PE1 I/O (port) | TIOC0B I/O (MTU2) | RXD0 input (SCI) | — | — |
| | PE2 I/O (port) | TIOC0C I/O (MTU2) | TXD0 output (SCI) | — | — |
| | PE3 I/O (port) | TIOC0D I/O (MTU2) | SCK0 I/O (SCI) | — | — |
| | PE4 I/O (port) | A11 output (BSC) | TIOC1A I/O (MTU2) | RXD1 input (SCI) | — |
| | PE5 I/O (port) | A12 output (BSC) | TIOC1B I/O (MTU2) | TXD1 output (SCI) | — |
| | PE6 I/O (port) | A13 output (BSC) | TIOC2A I/O (MTU2) | SCK1 I/O (SCI) | — |
| | PE7 I/O (port) | A14 output (BSC) | TIOC2B I/O (MTU2) | — | — |
| | PE8 I/O (port) | A15 output (BSC) | TIOC3A I/O (MTU2) | — | — |
| | PE9 I/O (port) | TIOC3B I/O (MTU2) | — | — | — |
| | PE10 I/O (port) | $\overline{\text{CS0}}$ output (BSC) | TIOC3C I/O (MTU2) | — | — |
| | PE11 I/O (port) | TIOC3D I/O (MTU2) | — | — | — |
| | PE12 I/O (port) | TIOC4A I/O (MTU2) | — | — | — |
| | PE13 I/O (port) | TIOC4B I/O (MTU2) | $\overline{\text{MRES}}$ input (INTC) | — | — |
| | PE14 I/O (port) | TIOC4C I/O (MTU2) | — | — | — |
| | PE15 I/O (port) | TIOC4D I/O (MTU2) | $\overline{\text{IRQOUT}}$ output (INTC) | — | — |
| | PE16 I/O (port) | $\overline{\text{WAIT}}$ input (BSC) | TIOC3BS I/O (MTU2S) | $\overline{\text{ASEBRKAK}}$ output (E10A)* | $\overline{\text{ASEBRK}}$ input (E10A)* |
| | PE17 I/O (port) | $\overline{\text{CS0}}$ output (BSC) | TIOC3DS I/O (MTU2S) | TCK input (H-UDI)* | — |
| | PE18 I/O (port) | $\overline{\text{CS1}}$ output (BSC) | TIOC4AS I/O (MTU2S) | TDI input (H-UDI)* | — |
| | PE19 I/O (port) | $\overline{\text{RD}}$ output (BSC) | TIOC4BS I/O (MTU2S) | TDO output (H-UDI)* | — |
| | PE20 I/O (port) | TIOC4CS I/O (MTU2S) | TMS input (H-UDI)* | — | — |
| | PE21 I/O (port) | $\overline{\text{WRL}}$ output (BSC) | TIOC4DS I/O (MTU2S) | $\overline{\text{TRST}}$ input (H-UDI)* | — |

Note: * Function enabled on the SH7137 only.

**Table 20.8 SH7131/SH7136 Multiplexed Pins (Port F)**

| Port | Function 1 (Related Module) | Function 2 (Related Module) |
|---|---|---|
| F | PF0 input (port) | AN0 input (A/D) |
| | PF1 input (port) | AN1 input (A/D) |
| | PF2 input (port) | AN2 input (A/D) |
| | PF3 input (port) | AN3 input (A/D) |
| | PF8 input (port) | AN8 input (A/D) |
| | PF9 input (port) | AN9 input (A/D) |
| | PF10 input (port) | AN10 input (A/D) |
| | PF11 input (port) | AN11 input (A/D) |
| | PF12input (port) | AN12 input (A/D) |
| | PF13 input (port) | AN13 input (A/D) |
| | PF14 input (port) | AN14 input (A/D) |
| | PF15 input (port) | AN15 input (A/D) |

Note: During A/D conversion, the AN input function is enabled.

**Table 20.9 SH7132/SH7137 Multiplexed Pins (Port F)**

| Port | Function 1 (Related Module) | Function 2 (Related Module) |
|---|---|---|
| F | PF0 input (port) | AN0 input (A/D) |
| | PF1 input (port) | AN1 input (A/D) |
| | PF2 input (port) | AN2 input (A/D) |
| | PF3 input (port) | AN3 input (A/D) |
| | PF4 input (port) | AN4 input (A/D) |
| | PF5 input (port) | AN5 input (A/D) |
| | PF6 input (port) | AN6 input (A/D) |
| | PF7 input (port) | AN7 input (A/D) |
| | PF8 input (port) | AN8 input (A/D) |
| | PF9 input (port) | AN9 input (A/D) |
| | PF10 input (port) | AN10 input (A/D) |
| | PF11 input (port) | AN11 input (A/D) |
| | PF12input (port) | AN12 input (A/D) |
| | PF13 input (port) | AN13 input (A/D) |
| | PF14 input (port) | AN14 input (A/D) |
| | PF15 input (port) | AN15 input (A/D) |

Note: During A/D conversion, the AN input function is enabled.

## Table 20.10 SH7131/SH7136 Pin Functions in Each Operating Mode

| | Pin Name | |
|---|---|---|
| | Single-Chip Mode (MCU Mode 3) | |
| **Pin No.** | **Initial Function** | **Functions Selectable by PFC** |
| 10, 27, 44, 57, 79 | Vcc | Vcc |
| 1, 13, 29, 49 | Vss | Vss |
| 15, 46 | $V_{CL}$ | $V_{CL}$ |
| 78 | AVcc | AVcc |
| 63 | AVss | AVss |
| 60 | PLLVss | PLLVss |
| 68 | AVrefh | AVrefh |
| 73 | AVrefl | AVrefl |
| 56 | EXTAL | EXTAL |
| 55 | XTAL | XTAL |
| 62 | MD1 | MD1 |
| 59 | FWE*[1] | FWE*[1] |
| 54 | $\overline{\text{RES}}$ | $\overline{\text{RES}}$ |
| 80 | $\overline{\text{WDTOVF}}$ | $\overline{\text{WDTOVF}}$ |
| 58 | NMI | NMI |
| 61 | $\overline{\text{ASEMD0}}$*[1] | $\overline{\text{ASEMD0}}$*[1] |
| 53 | PA0 | PA0/$\overline{\text{POE0}}$/RXD0 |
| 52 | PA1 | PA1/$\overline{\text{POE1}}$/TXD0 |
| 51 | PA2 | PA2/IRQ0/$\overline{\text{POE2}}$/SCK0 |
| 50 | PA3 | PA3/IRQ1/RXD1 |
| 48 | PA4 | PA4/IRQ2/TXD1 |
| 47 | PA5 | PA5/IRQ3/SCK1 |
| 45 | PA6 | PA6/$\overline{\text{UBCTRG}}$*[2]/TCLKA/$\overline{\text{POE4}}$ |
| 43 | PA7 | PA7/TCLKB/$\overline{\text{POE5}}$/SCK2 |
| 42 | PA8 | PA8/TCLKC/$\overline{\text{POE6}}$/RXD2 |
| 41 | PA9 | PA9/TCLKD/$\overline{\text{POE8}}$/TXD2 |
| 40 | PA10 | PA10/RXD0 |
| 39 | PA11 | PA11/TXD0/$\overline{\text{ADTRG}}$ |

| | Pin Name | |
|---|---|---|
| | Single-Chip Mode (MCU Mode 3) | |
| **Pin No.** | **Initial Function** | **Functions Selectable by PFC** |
| 38 | PA12 | PA12/SCK0/$\overline{\text{SCS}}$ |
| 37 | PA13 | PA13/SCK1/SSCK |
| 36 | PA14 | PA14/RXD1/SSI |
| 35 | PA15 | PA15/TXD1/SSO |
| 34 | PB2 | PB2/IRQ0/$\overline{\text{POE0}}$/TIC5VS/SCL |
| 33 | PB3 | PB3/IRQ1/$\overline{\text{POE1}}$/TIC5V/SDA |
| 32 | PB4 | PB4/IRQ2/$\overline{\text{POE4}}$/TIC5US |
| 31 | PB5 | PB5/IRQ3/$\overline{\text{POE5}}$/TIC5U |
| 30 | PB6 | PB6/CTx0 |
| 28 | PB7 | PB7/CRx0 |
| 26 | PE0 | PE0/TIOC0A |
| 25 | PE1 | PE1/TIOC0B/RXD0 |
| 24 | PE2 | PE2/TIOC0C/TXD0 |
| 23 | PE3 | PE3/TIOC0D/SCK0 |
| 22 | PE4 | PE4/TIOC1A/RXD1 |
| 21 | PE5 | PE5/TIOC1B/TXD1 |
| 20 | PE6 | PE6/TIOC2A/SCK1 |
| 19 | PE7 | PE7/TIOC2B |
| 18 | PE8 | PE8/TIOC3A |
| 16 | PE9 | PE9/TIOC3B |
| 17 | PE10 | PE10/TIOC3C |
| 14 | PE11 | PE11/TIOC3D |
| 12 | PE12 | PE12/TIOC4A |
| 11 | PE13 | PE13/TIOC4B/$\overline{\text{MRES}}$ |
| 9 | PE14 | PE14/TIOC4C |
| 8 | PE15 | PE15/TIOC4D/$\overline{\text{IRQOUT}}$ |
| 7 | PE16/($\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$*[1]) | PE16/TIOC3BS |
| 6 | PE17/(TCK*[1]) | PE17/TIOC3DS |
| 5 | PE18/(TDI*[1]) | PE18/TIOC4AS |

| | Pin Name | |
|---|---|---|
| | Single-Chip Mode (MCU Mode 3) | |
| **Pin No.** | **Initial Function** | **Functions Selectable by PFC** |
| 4 | PE19/(TDO*[1]) | PE19/TIOC4BS |
| 3 | PE20/(TMS*[1]) | PE20/TIOC4CS |
| 2 | PE21/($\overline{\text{TRST}}$*[1]) | PE21/TIOC4DS |
| 77 | PF0/AN0 | PF0/AN0 |
| 76 | PF1/AN1 | PF1/AN1 |
| 75 | PF2/AN2 | PF2/AN2 |
| 74 | PF3/AN3 | PF3/AN3 |
| 72 | PF8/AN8 | PF8/AN8 |
| 71 | PF9/AN9 | PF9/AN9 |
| 70 | PF10/AN10 | PF10/AN10 |
| 69 | PF11/AN11 | PF11/AN11 |
| 67 | PF12/AN12 | PF12/AN12 |
| 66 | PF13/AN13 | PF13/AN13 |
| 65 | PF14/AN14 | PF14/AN14 |
| 64 | PF15/AN15 | PF15/AN15 |

Notes: 1. Fixed to TMS, $\overline{\text{TRST}}$, TDI, TDO, TCK, and $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$ when using the E10A ($\overline{\text{ASEMD0}}$ = low).

2. Function enabled on the SH7136 only.

## Table 20.11 SH7132/SH7137 Pin Functions in Each Operating Mode (1)

| | Pin Name | |
|---|---|---|
| | On-Chip ROM Disabled (MCU Mode 0) | |
| **Pin No.** | **Initial Function** | **Functions Selectable by PFC** |
| 3, 11, 36, 48, 57, 99 | Vcc | Vcc |
| 1, 14, 39, 50, 64 | Vss | Vss |
| 16, 59 | $V_{CL}$ | $V_{CL}$ |
| 98 | AVcc | AVcc |
| 79 | AVss | AVss |
| 88 | AVrefh | AVrefh |
| 93 | AVrefl | AVrefl |
| 75 | PLLVss | PLLVss |
| 72 | EXTAL | EXTAL |
| 71 | XTAL | XTAL |
| 78 | MD0 | MD0 |
| 77 | MD1 | MD1 |
| 74 | FWE*[1] | FWE*[1] |
| 70 | $\overline{\text{RES}}$ | $\overline{\text{RES}}$ |
| 100 | $\overline{\text{WDTOVF}}$ | $\overline{\text{WDTOVF}}$ |
| 73 | NMI | NMI |
| 76 | $\overline{\text{ASEMD0}}$*[1] | $\overline{\text{ASEMD0}}$*[1] |
| 69 | A0 | PA0/A0/$\overline{\text{POE0}}$/RXD0 |
| 68 | A1 | PA1/A1/$\overline{\text{POE1}}$/TXD0 |
| 67 | A2 | PA2/A2/IRQ0/$\overline{\text{POE2}}$/SCK0 |
| 66 | A3 | PA3/A3/IRQ1/RXD1 |
| 65 | A4 | PA4/A4/IRQ2/TXD1 |
| 63 | A5 | PA5/A5/IRQ3/SCK1 |
| 62 | $\overline{\text{RD}}$ | PA6/$\overline{\text{RD}}$/$\overline{\text{UBCTRG}}$*[2]/TCLKA/$\overline{\text{POE4}}$ |
| 61 | PA7 | PA7/TCLKB/$\overline{\text{POE5}}$/SCK2 |
| 60 | $\overline{\text{WRL}}$ | PA8/$\overline{\text{WRL}}$/TCLKC/$\overline{\text{POE6}}$/RXD2 |
| 58 | PA9 | PA9/$\overline{\text{WAIT}}$/TCLKD/$\overline{\text{POE8}}$/TXD2 |
| 56 | A6 | PA10/A6/RXD0 |

| | Pin Name | |
|---|---|---|
| | On-Chip ROM Disabled (MCU Mode 0) | |
| **Pin No.** | **Initial Function** | **Functions Selectable by PFC** |
| 55 | A7 | PA11/A7/TXD0/$\overline{\text{ADTRG}}$ |
| 54 | A8 | PA12/A8/SCK0/$\overline{\text{SCS}}$ |
| 53 | A9 | PA13/A9/SCK1/SSCK |
| 52 | A10 | PA14/A10/RXD1/SSI |
| 51 | CK | PA15/CK/TXD1/SSO |
| 49 | PB0 | PB0/$\overline{\text{BACK}}$/TIC5WS |
| 47 | PB1 | PB1/$\overline{\text{BREQ}}$/TIC5W |
| 46 | A16 | PB2/A16/IRQ0/$\overline{\text{POE0}}$/TIC5VS/SCL |
| 45 | A17 | PB3/A17/IRQ1/$\overline{\text{POE1}}$/TIC5V/SDA |
| 44 | PB4 | PB4/A18/IRQ2/$\overline{\text{POE4}}$/TIC5US |
| 43 | PB5 | PB5/A19/IRQ3/$\overline{\text{POE5}}$/TIC5U |
| 42 | PB6 | PB6/$\overline{\text{WAIT}}$/CTx0 |
| 41 | PB7 | PB7/$\overline{\text{CS1}}$/CRx0 |
| 40 | D0 | PD0/D0/RXD0 |
| 38 | D1 | PD1/D1/TXD0 |
| 37 | D2 | PD2/D2/SCK0 |
| 35 | D3 | PD3/D3/RXD1 |
| 34 | D4 | PD4/D4/TXD1 |
| 33 | D5 | PD5/D5/SCK1 |
| 32 | D6 | PD6/D6/RXD2 |
| 31 | D7 | PD7/D7/TXD2/$\overline{\text{SCS}}$ |
| 30 | PD8 | PD8/SCK2/SSCK |
| 29 | PD9 | PD9/SSI |
| 28 | PD10 | PD10/SSO |

| | Pin Name | |
|---|---|---|
| | On-Chip ROM Disabled (MCU Mode 0) | |
| **Pin No.** | **Initial Function** | **Functions Selectable by PFC** |
| 27 | PE0 | PE0/TIOC0A |
| 26 | PE1 | PE1/TIOC0B/RXD0 |
| 25 | PE2 | PE2/TIOC0C/TXD0 |
| 24 | PE3 | PE3/TIOC0D/SCK0 |
| 23 | A11 | PE4/A11/TIOC1A/RXD1 |
| 22 | A12 | PE5/A12/TIOC1B/TXD1 |
| 21 | A13 | PE6/A13/TIOC2A/SCK1 |
| 20 | A14 | PE7/A14/TIOC2B |
| 19 | A15 | PE8/A15/TIOC3A |
| 17 | PE9 | PE9/TIOC3B |
| 18 | $\overline{CS0}$ | PE10/$\overline{CS0}$/TIOC3C |
| 15 | PE11 | PE11/TIOC3D |
| 13 | PE12 | PE12/TIOC4A |
| 12 | PE13 | PE13/TIOC4B/$\overline{MRES}$ |
| 10 | PE14 | PE14/TIOC4C |
| 9 | PE15 | PE15/TIOC4D/$\overline{IRQOUT}$ |
| 8 | PE16/($\overline{ASEBRKAK}$/$\overline{ASEBRK}$*[1]) | PE16/$\overline{WAIT}$/TIOC3BS |
| 7 | PE17/(TCK*[1]) | PE17/$\overline{CS0}$/TIOC3DS |
| 6 | PE18/(TDI*[1]) | PE18/$\overline{CS1}$/TIOC4AS |
| 5 | PE19/(TDO*[1]) | PE19/$\overline{RD}$/TIOC4BS |
| 4 | PE20/(TMS*[1]) | PE20/TIOC4CS |
| 2 | PE21/($\overline{TRST}$*[1]) | PE21/$\overline{WRL}$/TIOC4DS |
| 97 | PF0/AN0 | PF0/AN0 |
| 96 | PF1/AN1 | PF1/AN1 |
| 95 | PF2/AN2 | PF2/AN2 |
| 94 | PF3/AN3 | PF3/AN3 |
| 92 | PF4/AN4 | PF4/AN4 |
| 91 | PF5/AN5 | PF5/AN5 |
| 90 | PF6/AN6 | PF6/AN6 |

| | Pin Name | |
|---|---|---|
| | On-Chip ROM Disabled (MCU Mode 0) | |
| **Pin No.** | **Initial Function** | **Functions Selectable by PFC** |
| 89 | PF7/AN7 | PF7/AN7 |
| 87 | PF8/AN8 | PF8/AN8 |
| 86 | PF9/AN9 | PF9/AN9 |
| 85 | PF10/AN10 | PF10/AN10 |
| 84 | PF11/AN11 | PF11/AN11 |
| 83 | PF12/AN12 | PF12/AN12 |
| 82 | PF13/AN13 | PF13/AN13 |
| 81 | PF14/AN14 | PF14/AN14 |
| 80 | PF15/AN15 | PF15/AN15 |

Notes: 1. Fixed to TMS, $\overline{\text{TRST}}$, TDI, TDO, TCK, and $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$ when using the E10A ($\overline{\text{ASEMD0}}$ = low).

2. Function enabled on the SH7137 only.

## Table 20.12 SH7132/SH7137 Pin Functions in Each Operating Mode (2)

| | Pin Name | | | |
|---|---|---|---|---|
| | On-Chip ROM Enabled (MCU Mode 2) | | Single-Chip Mode (MCU Mode 3) | |
| Pin No. | Initial Function | Functions Selectable by PFC | Initial Function | Functions Selectable by PFC |
| 3, 11, 36, 48, 57, 99 | Vcc | Vcc | Vcc | Vcc |
| 1,14,39,50,64 | Vss | Vss | Vss | Vss |
| 16,59 | $V_{CL}$ | $V_{CL}$ | $V_{CL}$ | $V_{CL}$ |
| 98 | AVcc | AVcc | AVcc | AVcc |
| 79 | AVss | AVss | AVss | AVss |
| 88 | AVrefh | AVrefh | AVreth | AVreth |
| 93 | AVrefl | AVrefl | AVrefl | AVrefl |
| 75 | PLLVss | PLLVss | PLLVss | PLLVss |
| 72 | EXTAL | EXTAL | EXTAL | EXTAL |
| 71 | XTAL | XTAL | XTAL | XTAL |
| 78 | MD0 | MD0 | MD0 | MD0 |
| 77 | MD1 | MD1 | MD1 | MD1 |
| 74 | FWE | FWE | FWE | FWE |
| 70 | $\overline{RES}$ | $\overline{RES}$ | $\overline{RES}$ | $\overline{RES}$ |
| 100 | $\overline{WDTOVF}$ | $\overline{WDTOVF}$ | $\overline{WDTOVF}$ | $\overline{WDTOVF}$ |
| 73 | NMI | NMI | NMI | NMI |
| 76 | $\overline{ASEMD0}$ | $\overline{ASEMD0}$ | $\overline{ASEMD0}$ | $\overline{ASEMD0}$ |
| 69 | PA0 | PA0/A0/$\overline{POE0}$/RXD0 | PA0 | PA0/$\overline{POE0}$/RXD0 |
| 68 | PA1 | PA1/A1/$\overline{POE1}$/TXD0 | PA1 | PA1/$\overline{POE1}$/TXD0 |
| 67 | PA2 | PA2/A2/IRQ0/$\overline{POE2}$/SCK0 | PA2 | PA2/IRQ0/$\overline{POE2}$/SCK0 |
| 66 | PA3 | PA3/A3/IRQ1/RXD1 | PA3 | PA3/IRQ1/RXD1 |
| 65 | PA4 | PA4/A4/IRQ2/TXD1 | PA4 | PA4/IRQ2/TXD1 |
| 63 | PA5 | PA5/A5/IRQ3/SCK1 | PA5 | PA5/IRQ3/SCK1 |
| 62 | PA6 | PA6/$\overline{RD}$/$\overline{UBCTRG}$*2/TCLKA/$\overline{POE4}$ | PA6 | PA6/$\overline{UBCTRG}$*2/TCLKA/$\overline{POE4}$ |
| 61 | PA7 | PA7/TCLKB/$\overline{POE5}$/SCK2 | PA7 | PA7/TCLKB/$\overline{POE5}$/SCK2 |
| 60 | PA8 | PA8/$\overline{WRL}$/TCLKC/$\overline{POE6}$/RXD2 | PA8 | PA8/TCLKC/$\overline{POE6}$/RXD2 |

| | Pin Name | | | |
|---|---|---|---|---|
| | On-Chip ROM Enabled (MCU Mode 2) | | Single-Chip Mode (MCU Mode 3) | |
| Pin No. | Initial Function | Functions Selectable by PFC | Initial Function | Functions Selectable by PFC |
| 58 | PA9 | PA9/$\overline{\text{WAIT}}$/TCLKD/$\overline{\text{POE8}}$/TXD2 | PA9 | PA9/TCLKD/$\overline{\text{POE8}}$/TXD2 |
| 56 | PA10 | PA10/A6/RXD0 | PA10 | PA10/RXD0 |
| 55 | PA11 | PA11/A7/TXD0/$\overline{\text{ADTRG}}$ | PA11 | PA11/TXD0/$\overline{\text{ADTRG}}$ |
| 54 | PA12 | PA12/A8/SCK0/$\overline{\text{SCS}}$ | PA12 | PA12/SCK0/$\overline{\text{SCS}}$ |
| 53 | PA13 | PA13/A9/SCK1/SSCK | PA13 | PA13/SCK1/SSCK |
| 52 | PA14 | PA14/A10/RXD1/SSI | PA14 | PA14/RXD1/SSI |
| 51 | CK | PA15/CK/TXD1/SSO | PA15 | PA15/TXD1/SSO |
| 49 | PB0 | PB0/$\overline{\text{BACK}}$/TIC5WS | PB0 | PB0//TIC5WS |
| 47 | PB1 | PB1/$\overline{\text{BREQ}}$/TIC5W | PB1 | PB1//TIC5W |
| 46 | PB2 | PB2/A16/IRQ0/$\overline{\text{POE0}}$/TIC5VS/SCL | PB2 | PB2/IRQ0/$\overline{\text{POE0}}$/TIC5VS/SCL |
| 45 | PB3 | PB3/A17/IRQ1/$\overline{\text{POE1}}$/TIC5V/SDA | PB3 | PB3/IRQ1/$\overline{\text{POE1}}$/TIC5V/SDA |
| 44 | PB4 | PB4/A18/IRQ2/$\overline{\text{POE4}}$/TIC5US | PB4 | PB4/IRQ2/$\overline{\text{POE4}}$/TIC5US |
| 43 | PB5 | PB5/A19/IRQ3/$\overline{\text{POE5}}$/TIC5U | PB5 | PB5/IRQ3/$\overline{\text{POE5}}$/TIC5U |
| 42 | PB6 | PB6/$\overline{\text{WAIT}}$/CTx0 | PB6 | PB6/CTx0 |
| 41 | PB7 | PB7/$\overline{\text{CS1}}$/CRx0 | PB7 | PB7/CRx0 |
| 40 | PD0 | PD0/D0/RXD0 | PD0 | PD0/RXD0 |
| 38 | PD1 | PD1/D1/TXD0 | PD1 | PD1/TXD0 |
| 37 | PD2 | PD2/D2/SCK0 | PD2 | PD2/SCK0 |
| 35 | PD3 | PD3/D3/RXD1 | PD3 | PD3/RXD1 |
| 34 | PD4 | PD4/D4/TXD1 | PD4 | PD4/TXD1 |
| 33 | PD5 | PD5/D5/SCK1 | PD5 | PD5/SCK1 |
| 32 | PD6 | PD6/D6/RXD2 | PD6 | PD6/RXD2 |
| 31 | PD7 | PD7/D7/TXD2/$\overline{\text{SCS}}$ | PD7 | PD7/TXD2/$\overline{\text{SCS}}$ |
| 30 | PD8 | PD8/SCK2/SSCK | PD8 | PD8/SCK2/SSCK |
| 29 | PD9 | PD9/SSI | PD9 | PD9/SSI |
| 28 | PD10 | PD10/SSO | PD10 | PD10/SSO |

| | Pin Name | | | |
|---|---|---|---|---|
| | On-Chip ROM Enabled (MCU Mode 2) | | Single-Chip Mode (MCU Mode 3) | |
| Pin No. | Initial Function | Functions Selectable by PFC | Initial Function | Functions Selectable by PFC |
| 27 | PE0 | PE0/TIOC0A | PE0 | PE0/TIOC0A |
| 26 | PE1 | PE1/TIOC0B/RXD0 | PE1 | PE1/TIOC0B/RXD0 |
| 25 | PE2 | PE2/TIOC0C/TXD0 | PE2 | PE2/TIOC0C/TXD0 |
| 24 | PE3 | PE3/TIOC0D/SCK0 | PE3 | PE3/TIOC0D/SCK0 |
| 23 | PE4 | PE4/A11/TIOC1A/RXD1 | PE4 | PE4/TIOC1A/RXD1 |
| 22 | PE5 | PE5/A12/TIOC1B/TXD1 | PE5 | PE5/TIOC1B/TXD1 |
| 21 | PE6 | PE6/A13/TIOC2A/SCK1 | PE6 | PE6/TIOC2A/SCK1 |
| 20 | PE7 | PE7/A14/TIOC2B | PE7 | PE7/TIOC2B |
| 19 | PE8 | PE8/A15/TIOC3A | PE8 | PE8/TIOC3A |
| 17 | PE9 | PE9/TIOC3B | PE9 | PE9/TIOC3B |
| 18 | PE10 | PE10/$\overline{\text{CS0}}$/TIOC3C | PE10 | PE10/TIOC3C |
| 15 | PE11 | PE11/TIOC3D | PE11 | PE11/TIOC3D |
| 13 | PE12 | PE12/TIOC4A | PE12 | PE12/TIOC4A |
| 12 | PE13 | PE13/TIOC4B/$\overline{\text{MRES}}$ | PE13 | PE13/TIOC4B/$\overline{\text{MRES}}$ |
| 10 | PE14 | PE14/TIOC4C | PE14 | PE14/TIOC4C |
| 9 | PE15 | PE15/TIOC4D/$\overline{\text{IRQOUT}}$ | PE15 | PE15/TIOC4D/$\overline{\text{IRQOUT}}$ |
| 8 | PE16/($\overline{\text{ASEBRKAK}}$/ $\overline{\text{ASEBRK}}$*[1]) | PE16/$\overline{\text{WAIT}}$/TIOC3BS | PE16/($\overline{\text{ASEBRKAK}}$/ $\overline{\text{ASEBRK}}$*[1]) | PE16/TIOC3BS |
| 7 | PE17/(TCK*[1]) | PE17/$\overline{\text{CS0}}$/TIOC3DS | PE17/(TCK*[1]) | PE17/TIOC3DS |
| 6 | PE18/(TDI*[1]) | PE18/$\overline{\text{CS1}}$/TIOC4AS | PE18/(TDI*[1]) | PE18/TIOC4AS |
| 5 | PE19/(TDO*[1]) | PE19/$\overline{\text{RD}}$/TIOC4BS | PE19/(TDO*[1]) | PE19/TIOC4BS |
| 4 | PE20/(TMS*[1]) | PE20/TIOC4CS | PE20/(TMS*[1]) | PE20/TIOC4CS |
| 2 | PE21/($\overline{\text{TRST}}$*[1]) | PE21/$\overline{\text{WRL}}$/TIOC4DS | PE21/($\overline{\text{TRST}}$*[1]) | PE21/TIOC4DS |
| 97 | PF0/AN0 | PF0/AN0 | PF0/AN0 | PF0/AN0 |
| 96 | PF1/AN1 | PF1/AN1 | PF1/AN1 | PF1/AN1 |
| 95 | PF2/AN2 | PF2/AN2 | PF2/AN2 | PF2/AN2 |
| 94 | PF3/AN3 | PF3/AN3 | PF3/AN3 | PF3/AN3 |

| | Pin Name | | | |
|---|---|---|---|---|
| | On-Chip ROM Enabled (MCU Mode 2) | | Single-Chip Mode (MCU Mode 3) | |
| **Pin No.** | **Initial Function** | **Functions Selectable by PFC** | **Initial Function** | **Functions Selectable by PFC** |
| 92 | PF4/AN4 | PF4/AN4 | PF4/AN4 | PF4/AN4 |
| 91 | PF5/AN5 | PF5/AN5 | PF5/AN5 | PF5/AN5 |
| 90 | PF6/AN6 | PF6/AN6 | PF6/AN6 | PF6/AN6 |
| 89 | PF7/AN7 | PF7/AN7 | PF7/AN7 | PF7/AN7 |
| 87 | PF8/AN8 | PF8/AN8 | PF8/AN8 | PF8/AN8 |
| 86 | PF9/AN9 | PF9/AN9 | PF9/AN9 | PF9/AN9 |
| 85 | PF10/AN10 | PF10/AN10 | PF10/AN10 | PF10/AN10 |
| 84 | PF11/AN11 | PF11/AN11 | PF11/AN11 | PF11/AN11 |
| 83 | PF12/AN12 | PF12/AN12 | PF12/AN12 | PF12/AN12 |
| 82 | PF13/AN13 | PF13/AN13 | PF13/AN13 | PF13/AN13 |
| 81 | PF14/AN14 | PF14/AN14 | PF14/AN14 | PF14/AN14 |
| 80 | PF15/AN15 | PF15/AN15 | PF15/AN15 | PF15/AN15 |

Notes: 1. Fixed to TMS, $\overline{\text{TRST}}$, TDI, TDO, TCK, and $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$ when using the E10A ($\overline{\text{ASEMD0}}$ = low).

2. Function enabled on the SH7137 only.

## 20.1 Register Descriptions

The PFC has the following registers. For details on register addresses and register states in each processing state, refer to section 25, List of Registers.

**Table 20.13 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Port A I/O register L | PAIORL | R/W | H'0000 | H'FFFFD106 | 8, 16 |
| Port A control register L4 | PACRL4 | R/W | H'0000* | H'FFFFD110 | 8, 16, 32 |
| Port A control register L3 | PACRL3 | R/W | H'0000* | H'FFFFD112 | 8, 16 |
| Port A control register L2 | PACRL2 | R/W | H'0000* | H'FFFFD114 | 8, 16, 32 |
| Port A control register L1 | PACRL1 | R/W | H'0000* | H'FFFFD116 | 8, 16 |
| Port B I/O register L | PBIORL | R/W | H'0000 | H'FFFFD186 | 8, 16 |
| Port B control register L2 | PBCRL2 | R/W | H'0000 | H'FFFFD194 | 8, 16, 32 |
| Port B control register L1 | PBCRL1 | R/W | H'0000* | H'FFFFD196 | 8, 16 |
| Port D I/O register L | PDIORL | R/W | H'0000 | H'FFFFD286 | 8, 16 |
| Port D control register L3 | PDCRL3 | R/W | H'0000 | H'FFFFD292 | 8, 16 |
| Port D control register L2 | PDCRL2 | R/W | H'0000* | H'FFFFD294 | 8, 16, 32 |
| Port D control register L1 | PDCRL1 | R/W | H'0000* | H'FFFFD296 | 8, 16 |
| Port E I/O register H | PEIORH | R/W | H'0000 | H'FFFFD304 | 8, 16, 32 |
| Port E I/O register L | PEIORL | R/W | H'0000 | H'FFFFD306 | 8, 16 |
| Port E control register H2 | PECRH2 | R/W | H'0000 | H'FFFFD30C | 8, 16, 32 |
| Port E control register H1 | PECRH1 | R/W | H'0000 | H'FFFFD30E | 8, 16 |
| Port E control register L4 | PECRL4 | R/W | H'0000 | H'FFFFD310 | 8, 16, 32 |
| Port E control register L3 | PECRL3 | R/W | H'0000* | H'FFFFD312 | 8, 16 |
| Port E control register L2 | PECRL2 | R/W | H'0000* | H'FFFFD314 | 8, 16, 32 |
| Port E control register L1 | PECRL1 | R/W | H'0000 | H'FFFFD316 | 8, 16 |
| IRQOUT function control register | IFCR | R/W | H'0000 | H'FFFFD322 | 8, 16 |

Note: For SH7132 and SH7137, the initial value differs in the on-chip ROM enabled/disabled external-extension mode. For details, refer to register descriptions in this section.

### 20.1.1 Port A I/O Register L (PAIORL)

PAIORL is a 16-bit readable/writable register that is used to set the pins on port A as inputs or outputs. Bits PA15IOR to PA0IOR correspond to pins PA15 to PA0 (names of multiplexed pins are here given as port names and pin numbers alone). PAIORL is enabled when the port A pins are functioning as general-purpose inputs/outputs (PA15 to PA0). In other states, PAIORL is disabled.

A given pin on port A will be an output pin if the corresponding bit in PAIORL is set to 1, and an input pin if the bit is cleared to 0.

The initial value of PAIORL is H'0000.

| 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 |

### 20.1.2 Port A Control Registers L1 to L4 (PACRL1 to PACRL4)

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

**SH7131/SH7136:**

- Port A Control Register L4 (PACRL4)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA15 MD2 | PA15 MD1 | PA15 MD0 | - | PA14 MD2 | PA14 MD1 | PA14 MD0 | - | PA13 MD2 | PA13 MD1 | PA13 MD0 | - | PA12 MD2 | PA12 MD1 | PA12 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PA15MD2 | 0 | R/W | PA15 Mode |
| 13 | PA15MD1 | 0 | R/W | Select the function of the PA15/TXD1/SSO pin. |
| 12 | PA15MD0 | 0 | R/W | 000: PA15 I/O (port)<br>101: SSO I/O (SSU)<br>110: TXD1 output (SCI)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PA14MD2 | 0 | R/W | PA14 Mode |
| 9 | PA14MD1 | 0 | R/W | Select the function of the PA14/RXD1/SSI pin. |
| 8 | PA14MD0 | 0 | R/W | 000: PA14 I/O (port)<br>101: SSI I/O (SSU)<br>110: RXD1 input (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PA13MD2 | 0 | R/W | PA13 Mode |
| 5 | PA13MD1 | 0 | R/W | Select the function of the PA13/SCK1/SSCK pin. |
| 4 | PA13MD0 | 0 | R/W | 000: PA13 I/O (port)<br>101: SSCK I/O (SSU)<br>110: SCK1 I/O (SCI)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | PA12MD2 | 0 | R/W | PA12 Mode |
| 1 | PA12MD1 | 0 | R/W | Select the function of the PA12/SCK0/$\overline{\text{SCS}}$ pin. |
| 0 | PA12MD0 | 0 | R/W | 000: PA12 I/O (port) |
| | | | | 101: $\overline{\text{SCS}}$ I/O (SSU) |
| | | | | 110: SCK0 I/O (SCI) |
| | | | | Other than above: Setting prohibited |

- Port A Control Register L3 (PACRL3)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA11 MD2 | PA11 MD1 | PA11 MD0 | - | PA10 MD2 | PA10 MD1 | PA10 MD0 | - | PA9 MD2 | PA9 MD1 | PA9 MD0 | - | PA8 MD2 | PA8 MD1 | PA8 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PA11MD2 | 0 | R/W | PA11 Mode |
| 13 | PA11MD1 | 0 | R/W | Select the function of the PA11/TXD0/$\overline{\text{ADTRG}}$ pin. |
| 12 | PA11MD0 | 0 | R/W | 000: PA11 I/O (port)<br>010: $\overline{\text{ADTRG}}$ input (A/D)<br>110: TXD0 output (SCI)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PA10MD2 | 0 | R/W | PA10 Mode |
| 9 | PA10MD1 | 0 | R/W | Select the function of the PA10/RXD0 pin. |
| 8 | PA10MD0 | 0 | R/W | 000: PA10 I/O (port)<br>110: RXD0 input (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PA9MD2 | 0 | R/W | PA9 Mode |
| 5 | PA9MD1 | 0 | R/W | Select the function of the PA9/TCLKD/TXD2 pin. |
| 4 | PA9MD0 | 0 | R/W | 000: PA9 I/O (port)<br>001: TCLKD input (MTU2)<br>110: TXD2 output (SCI)<br>111: $\overline{\text{POE8}}$ input (POE)<br>Other than above: Setting prohibited |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2<br>1<br>0 | PA8MD2<br>PA8MD1<br>PA8MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PA8 Mode<br>Select the function of the PA8/TCLKC/$\overline{\text{POE6}}$/RXD2 pin.<br>000: PA8 I/O (port)<br>001: TCLKC input (MTU2)<br>110: RXD2 input (SCI)<br>111: $\overline{\text{POE6}}$ input (POE)<br>Other than above: Setting prohibited |

- Port A Control Register L2 (PACRL2)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA7 MD2 | PA7 MD1 | PA7 MD0 | - | PA6 MD2 | PA6 MD1 | PA6 MD0 | - | PA5 MD2 | PA5 MD1 | PA5 MD0 | - | PA4 MD2 | PA4 MD1 | PA4 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14<br>13<br>12 | PA7MD2<br>PA7MD1<br>PA7MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PA7 Mode<br>Select the function of the PA7/TCLKB/$\overline{\text{POE5}}$/SCK2 pin.<br>000: PA7 I/O (port)<br>001: TCLKB input (MTU2)<br>110: SCK2 I/O (SCI)<br>111: $\overline{\text{POE5}}$ input (POE)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 | PA6MD2 | 0 | R/W | PA6 Mode |
| 9 | PA6MD1 | 0 | R/W | Select the function of the |
| 8 | PA6MD0 | 0 | R/W | PA6/$\overline{\text{UBCTRG}}$/TCLKA/$\overline{\text{POE4}}$ pin. |
| | | | | 000: PA6 I/O (port) |
| | | | | 001: TCLKA input (MTU2) |
| | | | | 101: $\overline{\text{UBCTRG}}$ output (UBC)* |
| | | | | 111: $\overline{\text{POE4}}$ input (POE) |
| | | | | Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved |
| | | | | This bit is always read as 0. The write value should always be 0. |
| 6 | PA5MD2 | 0 | R/W | PA5 Mode |
| 5 | PA5MD1 | 0 | R/W | Select the function of the PA5/IRQ3/SCK1 pin. |
| 4 | PA5MD0 | 0 | R/W | 000: PA5 I/O (port) |
| | | | | 001: SCK1 I/O (SCI) |
| | | | | 111: IRQ3 input (INTC) |
| | | | | Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved |
| | | | | This bit is always read as 0. The write value should always be 0. |
| 2 | PA4MD2 | 0 | R/W | PA4 Mode |
| 1 | PA4MD1 | 0 | R/W | Select the function of the PA4/IRQ2/TXD1 pin. |
| 0 | PA4MD0 | 0 | R/W | 000: PA4 I/O (port) |
| | | | | 001: TXD1 output (SCI) |
| | | | | 111: IRQ2 input (INTC) |
| | | | | Other than above: Setting prohibited |

Note: * Function enabled on the SH7136 only. Do not use this setting on the SH7131.

- Port A Control Register L1 (PACRL1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA3 MD2 | PA3 MD1 | PA3 MD0 | - | PA2 MD2 | PA2 MD1 | PA2 MD0 | - | PA1 MD2 | PA1 MD1 | PA1 MD0 | - | PA0 MD2 | PA0 MD1 | PA0 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PA3MD2 | 0 | R/W | PA3 Mode |
| 13 | PA3MD1 | 0 | R/W | Select the function of the PA3/IRQ1/RXD1 pin. |
| 12 | PA3MD0 | 0 | R/W | 000: PA3 I/O (port)<br>001: RXD1 input (SCI)<br>111: IRQ1 input (INTC)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PA2MD2 | 0 | R/W | PA2 Mode |
| 9 | PA2MD1 | 0 | R/W | Select the function of the PA2/IRQ0/$\overline{\text{POE2}}$/SCK0 pin. |
| 8 | PA2MD0 | 0 | R/W | 000: PA2 I/O (port)<br>001: SCK0 I/O (SCI)<br>011: IRQ0 input (INTC)<br>111: $\overline{\text{POE2}}$ input (POE)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PA1MD2 | 0 | R/W | PA1 Mode |
| 5 | PA1MD1 | 0 | R/W | Select the function of the PA1/$\overline{\text{POE1}}$/TXD0 pin. |
| 4 | PA1MD0 | 0 | R/W | 000: PA1 I/O (port)<br>001: TXD0 output (SCI)<br>111: $\overline{\text{POE1}}$ input (POE)<br>Other than above: Setting prohibited |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PA0MD2 | 0 | R/W | PA0 Mode |
| 1 | PA0MD1 | 0 | R/W | Select the function of the PA0/$\overline{\text{POE0}}$/RXD0 pin. |
| 0 | PA0MD0 | 0 | R/W | 000: PA0 I/O (port)<br>001: RXD0 input (SCI)<br>111: $\overline{\text{POE0}}$ input (POE)<br>Other than above: Setting prohibited |

**SH7132/SH7137:**

- Port A Control Register L4 (PACRL4)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA15 MD2 | PA15 MD1 | PA15 MD0 | - | PA14 MD2 | PA14 MD1 | PA14 MD0 | - | PA13 MD2 | PA13 MD1 | PA13 MD0 | - | PA12 MD2 | PA12 MD1 | PA12 MD0 |
| Initial value: | 0 | 0 | 0 | 0*1 | 0 | 0*2 | 0 | 0 | 0 | 0*2 | 0 | 0 | 0 | 0*2 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Notes: 1. The initial value is 1 in the on-chip ROM enabled/disabled external-extension mode.
2. The initial value is 1 in the on-chip ROM disabled external-extension mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PA15MD2 | 0 | R/W | PA15 Mode |
| 13 | PA15MD1 | 0 | R/W | Select the function of the PA15/CK/TXD1/SSO pin. |
| 12 | PA15MD0 | 0*1 | R/W | 000: PA15 I/O (port)<br>001: CK output (CPG)*3<br>101: SSO I/O (SSU)<br>110: TXD1 output (SCI)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 | PA14MD2 | 0*[2] | R/W | PA14 Mode |
| 9 | PA14MD1 | 0 | R/W | Select the function of the PA14/A10/RXD1/SSI pin. |
| 8 | PA14MD0 | 0 | R/W | 000: PA14 I/O (port)<br>100: A10 output (BSC)*[3]<br>101: SSI I/O (SSU)<br>110: RXD1 input (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PA13MD2 | 0*[2] | R/W | PA13 Mode |
| 5 | PA13MD1 | 0 | R/W | Select the function of the PA13/A9/SCK1/SSCK pin. |
| 4 | PA13MD0 | 0 | R/W | 000: PA13 I/O (port)<br>100: A9 output (BSC)*[3]<br>101: SSCK I/O (SSU)<br>110: SCK1 I/O (SCI)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PA12MD2 | 0*[2] | R/W | PA12 Mode |
| 1 | PA12MD1 | 0 | R/W | Select the function of the PA12/A8/SCK0/$\overline{\text{SCS}}$ pin. |
| 0 | PA12MD0 | 0 | R/W | 000: PA12 I/O (port)<br>100: A8 output (BSC)*[3]<br>101: $\overline{\text{SCS}}$ I/O (SSU)<br>110: SCK0 I/O (SCI)<br>Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM enabled/disabled external-extension mode.
2. The initial value is 1 in the on-chip ROM disabled external-extension mode.
3. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

- Port A Control Register L3 (PACRL3)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA11 MD2 | PA11 MD1 | PA11 MD0 | - | PA10 MD2 | PA10 MD1 | PA10 MD0 | - | PA9 MD2 | PA9 MD1 | PA9 MD0 | - | PA8 MD2 | PA8 MD1 | PA8 MD0 |
| Initial value: | 0 | 0*1 | 0 | 0 | 0 | 0*1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0*1 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Note: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PA11MD2 | 0*1 | R/W | PA11 Mode |
| 13 | PA11MD1 | 0 | R/W | Select the function of the PA11/A7/TXD0/$\overline{\text{ADTRG}}$ pin. |
| 12 | PA11MD0 | 0 | R/W | 000: PA11 I/O (port)<br>010: $\overline{\text{ADTRG}}$ input (A/D)<br>100: A7 output (BSC)*2<br>110: TXD0 output (SCI)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PA10MD2 | 0*1 | R/W | PA10 Mode |
| 9 | PA10MD1 | 0 | R/W | Select the function of the PA10/A6/RXD0 pin. |
| 8 | PA10MD0 | 0 | R/W | 000: PA10 I/O (port)<br>100: A6 output (BSC)*2<br>110: RXD0 input (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | PA9MD2 | 0 | R/W | PA9 Mode |
| 5 | PA9MD1 | 0 | R/W | Select the function of the PA9/$\overline{\text{WAIT}}$/TCLKD/TXD2 pin. |
| 4 | PA9MD0 | 0 | R/W | 000: PA9 I/O (port)<br>001: TCLKD input (MTU2)<br>100: $\overline{\text{WAIT}}$ input (BSC)*2<br>110: TXD2 output (SCI)<br>111: $\overline{\text{POE8}}$ input (POE)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PA8MD2 | 0*1 | R/W | PA8 Mode |
| 1 | PA8MD1 | 0 | R/W | Select the function of the PA8/$\overline{\text{WRL}}$/TCLKC/$\overline{\text{POE6}}$/RXD2 pin. |
| 0 | PA8MD0 | 0 | R/W | 000: PA8 I/O (port)<br>001: TCLKC input (MTU2)<br>100: $\overline{\text{WRL}}$ output (BSC)*2<br>110: RXD2 input (SCI)<br>111: $\overline{\text{POE6}}$ input (POE)<br>Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

- Port A Control Register L2 (PACRL2)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA7 MD2 | PA7 MD1 | PA7 MD0 | - | PA6 MD2 | PA6 MD1 | PA6 MD0 | - | PA5 MD2 | PA5 MD1 | PA5 MD0 | - | PA4 MD2 | PA4 MD1 | PA4 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0*1 | 0*1 | 0 | 0*1 | 0 | 0 | 0 | 0*1 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Note: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14<br>13<br>12 | PA7MD2<br>PA7MD1<br>PA7MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PA7 Mode<br>Select the function of the PA7/TCLKB/$\overline{\text{POE5}}$/SCK2 pin.<br>000: PA7 I/O (port)<br>001: TCLKB input (MTU2)<br>110: SCK2 I/O (SCI)<br>111: $\overline{\text{POE5}}$ input (POE)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10<br>9<br>8 | PA6MD2<br>PA6MD1<br>PA6MD0 | 0<br>0*1<br>0*1 | R/W<br>R/W<br>R/W | PA6 Mode<br>Select the function of the PA6/$\overline{\text{RD}}$/$\overline{\text{UBCTRG}}$/TCLKA/$\overline{\text{POE4}}$ pin.<br>000: PA6 I/O (port)<br>001: TCLKA input (MTU2)<br>011: $\overline{\text{RD}}$ output (BSC)*2<br>101: $\overline{\text{UBCTRG}}$ output (UBC)*3<br>111: $\overline{\text{POE4}}$ input (POE)<br>Other than above: Setting prohibited |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PA5MD2 | 0*[1] | R/W | PA5 Mode |
| 5 | PA5MD1 | 0 | R/W | Select the function of the PA5/A5/IRQ3/SCK1 pin. |
| 4 | PA5MD0 | 0 | R/W | 000: PA5 I/O (port)<br>001: SCK1 I/O (SCI)<br>100: A5 output (BSC)*[2]<br>111: IRQ3 input (INTC)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PA4MD2 | 0*[1] | R/W | PA4 Mode |
| 1 | PA4MD1 | 0 | R/W | Select the function of the PA4/A4/IRQ2/TXD1 pin. |
| 0 | PA4MD0 | 0 | R/W | 000: PA4 I/O (port)<br>001: TXD1 output (SCI)<br>100: A4 output (BSC)*[2]<br>111: IRQ2 input (INTC)<br>Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.
3. Function enabled on the SH7137 only. Do not use this setting on the SH7132.

- Port A Control Register L1 (PACRL1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA3 MD2 | PA3 MD1 | PA3 MD0 | - | PA2 MD2 | PA2 MD1 | PA2 MD0 | - | PA1 MD2 | PA1 MD1 | PA1 MD0 | - | PA0 MD2 | PA0 MD1 | PA0 MD0 |
| Initial value: | 0 | 0*1 | 0 | 0 | 0 | 0*1 | 0 | 0 | 0 | 0*1 | 0 | 0 | 0 | 0*1 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Note: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PA3MD2 | 0*1 | R/W | PA3 Mode |
| 13 | PA3MD1 | 0 | R/W | Select the function of the PA3/A3/IRQ1/RXD1 pin. |
| 12 | PA3MD0 | 0 | R/W | 000: PA3 I/O (port)<br>001: RXD1 input (SCI)<br>100: A3 output (BSC)*2<br>111: IRQ1 input (INTC)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PA2MD2 | 0*1 | R/W | PA2 Mode |
| 9 | PA2MD1 | 0 | R/W | Select the function of the PA2/A2/IRQ0/$\overline{\text{POE2}}$/SCK0 pin. |
| 8 | PA2MD0 | 0 | R/W | 000: PA2 I/O (port)<br>001: SCK0 I/O (SCI)<br>011: IRQ0 input (INTC)<br>100: A2 output (BSC)*2<br>111: $\overline{\text{POE2}}$ input (POE)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | PA1MD2 | 0*[1] | R/W | PA1 Mode |
| 5 | PA1MD1 | 0 | R/W | Select the function of the PA1/A1/$\overline{\text{POE1}}$/TXD0 pin. |
| 4 | PA1MD0 | 0 | R/W | 000: PA1 I/O (port)<br>001: TXD0 output (SCI)<br>100: A1 output (BSC)*[2]<br>111: $\overline{\text{POE1}}$ input (POE)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PA0MD2 | 0*[1] | R/W | PA0 Mode |
| 1 | PA0MD1 | 0 | R/W | Select the function of the PA0/A0/$\overline{\text{POE0}}$/RXD0 pin. |
| 0 | PA0MD0 | 0 | R/W | 000: PA0 I/O (port)<br>001: RXD0 input (SCI)<br>100: A0 output (BSC)*[2]<br>111: $\overline{\text{POE0}}$ input (POE)<br>Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

### 20.1.3 Port B I/O Register L (PBIORL)

PBIORL is a 16-bit readable/writable register that is used to set the pins on port B as inputs or outputs. Bits PB7IOR to PB0IOR correspond to pins PB7 to PB0, respectively (names of multiplexed pins are here given as port names and pin numbers alone). PBIORL is enabled when the port B pins are functioning as general-purpose inputs/outputs (PB7 to PB0). In other states, PBIORL is disabled.

A given pin on port B will be an output pin if the corresponding bit in PBIORL is set to 1, and an input pin if the bit is cleared to 0.

However, bits 1 and 0 of PBIORL are disabled in SH7131 and SH7136.

Bits 15 to 8 of PBIORL are reserved. These bits are always read as 0. The write value should always be 0.

The initial value of PBIORL is H'0000.

- Port B I/O Register L (PBIORL)

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

### 20.1.4 Port B Control Registers L1, L2 (PBCRL1, PBCRL2)

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

**SH7131/SH7136:**

- Port B Control Register L2 (PBCRL2)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PB7 MD2 | PB7 MD1 | PB7 MD0 | - | PB6 MD2 | PB6 MD1 | PB6 MD0 | - | PB5 MD2 | PB5 MD1 | PB5 MD0 | - | PB4 MD2 | PB4 MD1 | PB4 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PB7MD2 | 0 | R/W | PB7 Mode |
| 13 | PB7MD1 | 0 | R/W | Select the function of the PB7/CRx0 pin. |
| 12 | PB7MD0 | 0 | R/W | 000: PB7 I/O (port)<br>110: CRx0 input (RCAN-ET)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PB6MD2 | 0 | R/W | PB6 Mode |
| 9 | PB6MD1 | 0 | R/W | Select the function of the PB6/CTx0 pin. |
| 8 | PB6MD0 | 0 | R/W | 000: PB6 I/O (port)<br>110: CTx0 output (RCAN-ET)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PB5MD2 | 0 | R/W | PB5 Mode |
| 5 | PB5MD1 | 0 | R/W | Select the function of the PB5/IRQ3/$\overline{\text{POE5}}$/TIC5U pin. |
| 4 | PB5MD0 | 0 | R/W | 000: PB5 I/O (port)<br>001: IRQ3 input (INTC)<br>011: TIC5U input (MTU2)<br>111: $\overline{\text{POE5}}$ input (POE)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | PB4MD2 | 0 | R/W | PB4 Mode |
| 1 | PB4MD1 | 0 | R/W | Select the function of the PB4/IRQ2/$\overline{\text{POE4}}$/TIC5US pin. |
| 0 | PB4MD0 | 0 | R/W | 000: PB4 I/O (port) |
| | | | | 001: IRQ2 input (INTC) |
| | | | | 011: TIC5US input (MTU2S) |
| | | | | 111: $\overline{\text{POE4}}$ input (POE) |
| | | | | Other than above: Setting prohibited |

- Port B Control Register L1 (PBCRL1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PB3 MD2 | PB3 MD1 | PB3 MD0 | - | PB2 MD2 | PB2 MD1 | PB2 MD0 | - | - | - | - | - | - | - | - |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PB3MD2 | 0 | R/W | PB3 Mode |
| 13 | PB3MD1 | 0 | R/W | Select the function of the PB3/IRQ1/$\overline{\text{POE1}}$/TIC5V/SDA pin. |
| 12 | PB3MD0 | 0 | R/W | 000: PB3 I/O (port) |
| | | | | 001: IRQ1 input (INTC) |
| | | | | 010: $\overline{\text{POE1}}$ input (POE) |
| | | | | 011: TIC5V input (MTU2) |
| | | | | 100: SDA I/O (IIC2) |
| | | | | Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 | PB2MD2 | 0 | R/W | PB2 Mode |
| 9 | PB2MD1 | 0 | R/W | Select the function of the PB2/IRQ0/$\overline{\text{POE0}}$/TIC5VS/SCL pin. |
| 8 | PB2MD0 | 0 | R/W | 000: PB2 I/O (port)<br>001: IRQ0 input (INTC)<br>010: $\overline{\text{POE0}}$ input (POE)<br>011: TIC5VS input (MTU2S)<br>100: SCL I/O (IIC2)<br>Other than above: Setting prohibited |
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

**SH7132/SH7137:**

- Port B Control Register L2 (PBCRL2)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PB7 MD2 | PB7 MD1 | PB7 MD0 | - | PB6 MD2 | PB6 MD1 | PB6 MD0 | - | PB5 MD2 | PB5 MD1 | PB5 MD0 | - | PB4 MD2 | PB4 MD1 | PB4 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PB7MD2 | 0 | R/W | PB7 Mode |
| 13 | PB7MD1 | 0 | R/W | Select the function of the PB7/$\overline{\text{CS1}}$/CRx0 pin. |
| 12 | PB7MD0 | 0 | R/W | 000: PB7 I/O (port)<br>101: $\overline{\text{CS1}}$ output (BSC)*<br>110: CRx0 input (RCAN-ET)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PB6MD2 | 0 | R/W | PB6 Mode |
| 9 | PB6MD1 | 0 | R/W | Select the function of the PB6/$\overline{\text{WAIT}}$/CTx0 pin. |
| 8 | PB6MD0 | 0 | R/W | 000: PB6 I/O (port)<br>101: $\overline{\text{WAIT}}$ input (BSC)*<br>110: CTx0 output (RCAN-ET)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | PB5MD2 | 0 | R/W | PB5 Mode |
| 5 | PB5MD1 | 0 | R/W | Select the function of the PB5/A19/IRQ3/$\overline{\text{POE5}}$/TIC5U pin. |
| 4 | PB5MD0 | 0 | R/W | 000: PB5 I/O (port)<br>001: IRQ3 input (INTC)<br>011: TIC5U input (MTU2)<br>101: A19 output (BSC)*<br>111: $\overline{\text{POE5}}$ input (POE)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PB4MD2 | 0 | R/W | PB4 Mode |
| 1 | PB4MD1 | 0 | R/W | Select the function of the PB4/A18/IRQ2/$\overline{\text{POE4}}$/TIC5US pin. |
| 0 | PB4MD0 | 0 | R/W | 000: PB4 I/O (port)<br>001: IRQ2 input (INTC)<br>011: TIC5US input (MTU2S)<br>101: A18 output (BSC)*<br>111: $\overline{\text{POE4}}$ input (POE)<br>Other than above: Setting prohibited |

Note: * This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

- Port B Control Register L1 (PBCRL1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PB3 MD2 | PB3 MD1 | PB3 MD0 | - | PB2 MD2 | PB2 MD1 | PB2 MD0 | - | PB1 MD2 | PB1 MD1 | PB1 MD0 | - | PB0 MD2 | PB0 MD1 | PB0 MD0 |
| Initial value: | 0 | 0*1 | 0 | 0*1 | 0 | 0*1 | 0 | 0*1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Note: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14<br>13<br>12 | PB3MD2<br>PB3MD1<br>PB3MD0 | 0*1<br>0<br>0*1 | R/W<br>R/W<br>R/W | PB3 Mode<br>Select the function of the PB3/A17/IRQ1/$\overline{\text{POE1}}$/TIC5V/SDA pin.<br>000: PB3 I/O (port)<br>001: IRQ1 input (INTC)<br>010: $\overline{\text{POE1}}$ input (POE)<br>011: TIC5V input (MTU2)<br>100: SDA I/O (IIC2)<br>101: A17 output (BSC)*2<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10<br>9<br>8 | PB2MD2<br>PB2MD1<br>PB2MD0 | 0*1<br>0<br>0*1 | R/W<br>R/W<br>R/W | PB2 Mode<br>Select the function of the PB2/A16/IRQ0/$\overline{\text{POE0}}$/TIC5VS/SCL pin.<br>000: PB2 I/O (port)<br>001: IRQ0 input (INTC)<br>010: $\overline{\text{POE0}}$ input (POE)<br>011: TIC5VS input (MTU2S)<br>100: SCL I/O (IIC2)<br>101: A16 output (BSC)*2<br>Other than above: Setting prohibited |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PB1MD2 | 0 | R/W | PB1 Mode |
| 5 | PB1MD1 | 0 | R/W | Select the function of the PB1/$\overline{\text{BREQ}}$/TIC5W pin. |
| 4 | PB1MD0 | 0 | R/W | 000: PB1 I/O (port)<br>011: TIC5W input (MTU2)<br>101: $\overline{\text{BREQ}}$ input (BSC)*[2]<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PB0MD2 | 0 | R/W | PB0 Mode |
| 1 | PB0MD1 | 0 | R/W | Select the function of the PB0/$\overline{\text{BACK}}$/TIC5WS pin. |
| 0 | PB0MD0 | 0 | R/W | 000: PB0 I/O (port)<br>011: TIC5WS input (MTU2S)<br>101: $\overline{\text{BACK}}$ output (BSC)*[2]<br>Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

### 20.1.5 Port D I/O Register L (PDIORL) (SH7132 and SH7137 only)

PDIORL is a 16-bit readable/writable register that is used to set the pins on port D as inputs or outputs. Bits PD10IOR to PD0IOR correspond to pins PD10 to PD0 (names of multiplexed pins are here given as port names and pin numbers alone). PDIORL is enabled when the port D pins are functioning as general-purpose inputs/outputs (PD10 to PD0). In other states, PDIORL is disabled.

A given pin on port D will be an output pin if the corresponding bit in PDIORL is set to 1, and an input pin if the bit is cleared to 0.

However, PDIORL is disabled in SH7131 and SH7136.

Bits 15 to 11 of PDIORL are reserved. These bits are always read as 0. The write value should always be 0.

The initial value of PDIORL is H'0000.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | PD10 IOR | PD9 IOR | PD8 IOR | PD7 IOR | PD6 IOR | PD5 IOR | PD4 IOR | PD3 IOR | PD2 IOR | PD1 IOR | PD0 IOR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | 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 |

### 20.1.6 Port D Control Registers L1 to L3 (PDCRL1 to PDCRL3) (SH7132 and SH7137 only)

PDCRL1 to PDCRL3 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port D. However, PDCRL1 to PDCRL3 are disabled in SH7131 and SH7136.

- Port D Control Register L3 (PDCRL3)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | PD10 MD2 | PD10 MD1 | PD10 MD0 | - | PD9 MD2 | PD9 MD1 | PD9 MD0 | - | PD8 MD2 | PD8 MD1 | PD8 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 10 | PD10MD2 | 0 | R/W | PD10 Mode |
| 9 | PD10MD1 | 0 | R/W | Select the function of the PD10/SSO pin. |
| 8 | PD10MD0 | 0 | R/W | 000: PD10 I/O (port)<br>101: SSO I/O (SSU)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | PD9MD2 | 0 | R/W | PD9 Mode |
| 5 | PD9MD1 | 0 | R/W | Select the function of the PD9/SSI pin. |
| 4 | PD9MD0 | 0 | R/W | 000: PD9 I/O (port)<br>101: SSI I/O (SSU)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PD8MD2 | 0 | R/W | PD8 Mode |
| 1 | PD8MD1 | 0 | R/W | Select the function of the PD8/SCK2/SSCK pin. |
| 0 | PD8MD0 | 0 | R/W | 000: PD8 I/O (port)<br>101: SSCK I/O (SSU)<br>110: SCK2 I/O (SCI)<br>Other than above: Setting prohibited |

- Port D Control Register L2 (PDCRL2)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PD7 MD2 | PD7 MD1 | PD7 MD0 | - | PD6 MD2 | PD6 MD1 | PD6 MD0 | - | PD5 MD2 | PD5 MD1 | PD5 MD0 | - | PD4 MD2 | PD4 MD1 | PD4 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/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Note: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PD7MD2 | 0 | R/W | PD7 Mode |
| 13 | PD7MD1 | 0 | R/W | Select the function of the PD7/D7/$\overline{\text{SCS}}$/TXD2 pin. |
| 12 | PD7MD0 | 0*1 | R/W | 000: PD7 I/O (port)<br>001: D7 I/O (BSC)*2<br>101: $\overline{\text{SCS}}$ I/O (SSU)<br>110: TXD2 output (SCI)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PD6MD2 | 0 | R/W | PD6 Mode |
| 9 | PD6MD1 | 0 | R/W | Select the function of the PD6/D6/RXD2 pin. |
| 8 | PD6MD0 | 0*1 | R/W | 000: PD6 I/O (port)<br>001: D6 I/O (BSC)*2<br>110: RXD2 input (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | PD5MD2 | 0 | R/W | PD5 Mode |
| 5 | PD5MD1 | 0 | R/W | Select the function of the PD5/D5/SCK1 pin. |
| 4 | PD5MD0 | 0*[1] | R/W | 000: PD5 I/O (port)<br>001: D5 I/O (BSC)*[2]<br>110: SCK1 I/O (SCI)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PD4MD2 | 0 | R/W | PD4 Mode |
| 1 | PD4MD1 | 0 | R/W | Select the function of the PD4/D4/TXD1 pin. |
| 0 | PD4MD0 | 0*[1] | R/W | 000: PD4 I/O (port)<br>001: D4 I/O (BSC)*[2]<br>110: TXD1 output (SCI)<br>Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

- Port D Control Register L1 (PDCRL1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PD3 MD2 | PD3 MD1 | PD3 MD0 | - | PD2 MD2 | PD2 MD1 | PD2 MD0 | - | PD1 MD2 | PD1 MD1 | PD1 MD0 | - | PD0 MD2 | PD0 MD1 | PD0 MD0 |
| Initial value: | 0 | 0 | 0 | 0* | 0 | 0 | 0 | 0* | 0 | 0 | 0 | 0* | 0 | 0 | 0 | 0* |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Note: * The initial value is 1 in the on-chip ROM disabled external-extension mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 14 | PD3MD2 | 0 | R/W | PD3 Mode |
| 13 | PD3MD1 | 0 | R/W | Select the function of the PD3/D3/RXD1 pin. |
| 12 | PD3MD0 | 0*[1] | R/W | 000: PD3 I/O (port)<br>001: D3 I/O (BSC)*[2]<br>110: RXD1 input (SCI)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PD2MD2 | 0 | R/W | PD2 Mode |
| 9 | PD2MD1 | 0 | R/W | Select the function of the PD2/D2/SCK0 pin. |
| 8 | PD2MD0 | 0*[1] | R/W | 000: PD2 I/O (port)<br>001: D2 I/O (BSC)*[2]<br>110: SCK0 I/O (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PD1MD2 | 0 | R/W | PD1 Mode |
| 5 | PD1MD1 | 0 | R/W | Select the function of the PD1/D1/TXD0 pin. |
| 4 | PD1MD0 | 0*[1] | R/W | 000: PD1 I/O (port)<br>001: D1 I/O (BSC)*[2]<br>110: TXD0 output (SCI)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PD0MD2 | 0 | R/W | PD0 Mode |
| 1 | PD0MD1 | 0 | R/W | Select the function of the PD0/D0/RXD0 pin. |
| 0 | PD0MD0 | 0*[1] | R/W | 000: PD0 I/O (port)<br>001: D0 I/O (BSC)*[2]<br>110: RXD0 input (SCI)<br>Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

### 20.1.7 Port E I/O Registers L, H (PEIORL, PEIORH)

PEIORL and PEIORH are 16-bit readable/writable registers that are used to set the pins on port E as inputs or outputs. PE21IOR to PE0IOR correspond to pins PE21 to PE0 (names of multiplexed pins are here given as port names and pin numbers alone). PEIORL is enabled when the port E pins are functioning as general-purpose inputs/outputs (PE15 to PE0), and the TIOC pin is functioning as inputs/outputs of MTU2. In other states, PEIORL is disabled. PEIORH is enabled when the port E pins are functioning as general-purpose inputs/outputs (PE21 to PE16), and the TIOC pin is functioning as inputs/outputs of MTU2S. In other states, PEIORH is disabled.

A given pin on port E will be an output pin if the corresponding bit in PEIORH or PEIORL is set to 1, and an input pin if the bit is cleared to 0.

Bits 15 to 6 of PEIORH are reserved. These bits are always read as 0. The write value should always be 0.

The initial values of PEIORL and PEIORH are H'0000, respectively.

- Port E I/O Register H (PEIORH)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | PE21 IOR | PE20 IOR | PE19 IOR | PE18 IOR | PE17 IOR | PE16 IOR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

- Port E I/O Register L (PEIORL)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PE15 IOR | PE14 IOR | PE13 IOR | PE12 IOR | PE11 IOR | PE10 IOR | PE9 IOR | PE8 IOR | PE7 IOR | PE6 IOR | PE5 IOR | PE4 IOR | PE3 IOR | PE2 IOR | PE1 IOR | PE0 IOR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

### 20.1.8 Port E Control Registers L1 to L4, H1, H2 (PECRL1 to PECRL4, PECRH1, PECRH2)

PECRL1 to PECRL4, PECRH1 and PECRH2 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port E.

**SH7131/SH7136:**

- Port E Control Register H2 (PECRH2)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | PE21 MD1 | PE21 MD0 | - | - | PE20 MD1 | PE20 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5 | PE21MD1 | 0 | R/W | PE21 Mode |
| 4 | PE21MD0 | 0 | R/W | Select the function of the PE21/TIOC4DS/$\overline{\text{TRST}}$ pin. Fixed to $\overline{\text{TRST}}$ input when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE21 I/O (port)<br>01: TIOC4DS I/O (MTU2S)<br>Other than above: Setting prohibited |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | PE20MD1 | 0 | R/W | PE20 Mode |
| 0 | PE20MD0 | 0 | R/W | Select the function of the PE20/TIOC4CS/TMS pin. Fixed to TMS input when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE20 I/O (port)<br>01: TIOC4CS I/O (MTU2S)<br>Other than above: Setting prohibited |

- Port E Control Register H1 (PECRH1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | PE19 MD1 | PE19 MD0 | - | - | PE18 MD1 | PE18 MD0 | - | - | PE17 MD1 | PE17 MD0 | - | PE16 MD2 | PE16 MD1 | PE16 MD0 |
| 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/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15, 14 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 13 | PE19MD1 | 0 | R/W | PE19 Mode |
| 12 | PE19MD0 | 0 | R/W | Select the function of the PE19/TIOC4BS/TDO pin. Fixed to TDO output when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE19 I/O (port)<br>01: TIOC4BS I/O (MTU2S)<br>Other than above: Setting prohibited |
| 11, 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 9 | PE18MD1 | 0 | R/W | PE18 Mode |
| 8 | PE18MD0 | 0 | R/W | Select the function of the PE18/TIOC4AS/TDI pin. Fixed to TDI input when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE18 I/O (port)<br>01: TIOC4AS I/O (MTU2S)<br>Other than above: Setting prohibited |
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | PE17MD1 | 0 | R/W | PE17 Mode |
| 4 | PE17MD0 | 0 | R/W | Select the function of the PE17/TIOC3DS/TCK pin. Fixed to TCK input when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE17 I/O (port)<br>01: TIOC3DS I/O (MTU2S)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PE16MD2 | 0 | R/W | PE16 Mode |
| 1 | PE16MD1 | 0 | R/W | Select the function of the PE16/TIOC3BS/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$ pin. Fixed to $\overline{\text{ASEBRKAK}}$ output/$\overline{\text{ASEBRK}}$ input when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>000: PE16 I/O (port)<br>001: TIOC3BS I/O (MTU2S)<br>Other than above: Setting prohibited |
| 0 | PE16MD0 | 0 | R/W | |

- Port E Control Register L4 (PECRL4)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE15 MD2 | PE15 MD1 | PE15 MD0 | - | PE14 MD2 | PE14 MD1 | PE14 MD0 | - | - | PE13 MD1 | PE13 MD0 | - | PE12 MD2 | PE12 MD1 | PE12 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 14 | PE15MD2 | 0 | R/W | PE15 Mode |
| 13 | PE15MD1 | 0 | R/W | Select the function of the PE15/TIOC4D/$\overline{\text{IRQOUT}}$ pin. |
| 12 | PE15MD0 | 0 | R/W | 000: PE15 I/O (port)<br>001: TIOC4D I/O (MTU2)<br>011: $\overline{\text{IRQOUT}}$ output (INTC)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PE14MD2 | 0 | R/W | PE14 Mode |
| 9 | PE14MD1 | 0 | R/W | Select the function of the PE14/TIOC4C pin. |
| 8 | PE14MD0 | 0 | R/W | 000: PE14 I/O (port)<br>001: TIOC4C I/O (MTU2)<br>Other than above: Setting prohibited |
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5 | PE13MD1 | 0 | R/W | PE13 Mode |
| 4 | PE13MD0 | 0 | R/W | Select the function of the PE13/TIOC4B/$\overline{\text{MRES}}$ pin.<br>00: PE13 I/O (port)<br>01: TIOC4B I/O (MTU2)<br>10: $\overline{\text{MRES}}$ input (INTC)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PE12MD2 | 0 | R/W | PE12 Mode |
| 1 | PE12MD1 | 0 | R/W | Select the function of the PE12/TIOC4A pin. |
| 0 | PE12MD0 | 0 | R/W | 000: PE12 I/O (port)<br>001: TIOC4A I/O (MTU2)<br>Other than above: Setting prohibited |

- Port E Control Register L3 (PECRL3)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE11 MD2 | PE11 MD1 | PE11 MD0 | - | PE10 MD2 | PE10 MD1 | PE10 MD0 | - | PE9 MD2 | PE9 MD1 | PE9 MD0 | - | PE8 MD2 | PE8 MD1 | PE8 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PE11MD2 | 0 | R/W | PE11 Mode |
| 13 | PE11MD1 | 0 | R/W | Select the function of the PE11/TIOC3D pin. |
| 12 | PE11MD0 | 0 | R/W | 000: PE11 I/O (port)<br>001: TIOC3D I/O (MTU2)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PE10MD2 | 0 | R/W | PE10 Mode |
| 9 | PE10MD1 | 0 | R/W | Select the function of the PE10/TIOC3C pin. |
| 8 | PE10MD0 | 0 | R/W | 000: PE10 I/O (port)<br>001: TIOC3C I/O (MTU2)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PE9MD2 | 0 | R/W | PE9 Mode |
| 5 | PE9MD1 | 0 | R/W | Select the function of the PE9/TIOC3B pin. |
| 4 | PE9MD0 | 0 | R/W | 000: PE9 I/O (port)<br>001: TIOC3B I/O (MTU2)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | PE8MD2 | 0 | R/W | PE8 Mode |
| 1 | PE8MD1 | 0 | R/W | Select the function of the PE8/TIOC3A pin. |
| 0 | PE8MD0 | 0 | R/W | 000: PE8 I/O (port)<br>001: TIOC3A I/O (MTU2)<br>Other than above: Setting prohibited |

- Port E Control Register L2 (PECRL2)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE7 MD2 | PE7 MD1 | PE7 MD0 | - | PE6 MD2 | PE6 MD1 | PE6 MD0 | - | PE5 MD2 | PE5 MD1 | PE5 MD0 | - | PE4 MD2 | PE4 MD1 | PE4 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PE7MD2 | 0 | R/W | PE7 Mode |
| 13 | PE7MD1 | 0 | R/W | Select the function of the PE7/TIOC2B pin. |
| 12 | PE7MD0 | 0 | R/W | 000: PE7 I/O (port)<br>001: TIOC2B I/O (MTU2)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PE6MD2 | 0 | R/W | PE6 Mode |
| 9 | PE6MD1 | 0 | R/W | Select the function of the PE6/TIOC2A/SCK1 pin. |
| 8 | PE6MD0 | 0 | R/W | 000: PE6 I/O (port)<br>001: TIOC2A I/O (MTU2)<br>110: SCK1 I/O (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | PE5MD2 | 0 | R/W | PE5 Mode |
| 5 | PE5MD1 | 0 | R/W | Select the function of the PE5/TIOC1B/TXD1 pin. |
| 4 | PE5MD0 | 0 | R/W | 000: PE5 I/O (port)<br>001: TIOC1B I/O (MTU2)<br>110: TXD1 output (SCI)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PE4MD2 | 0 | R/W | PE4 Mode |
| 1 | PE4MD1 | 0 | R/W | Select the function of the PE4/TIOC1A/RXD1 pin. |
| 0 | PE4MD0 | 0 | R/W | 000: PE4 I/O (port)<br>001: TIOC1A I/O (MTU2)<br>110: RXD1 input (SCI)<br>Other than above: Setting prohibited |

- Port E Control Register L1 (PECRL1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE3 MD2 | PE3 MD1 | PE3 MD0 | - | PE2 MD2 | PE2 MD1 | PE2 MD0 | - | PE1 MD2 | PE1 MD1 | PE1 MD0 | - | - | PE0 MD1 | PE0 MD0 |
| 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<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PE3MD2 | 0 | R/W | PE3 Mode |
| 13 | PE3MD1 | 0 | R/W | Select the function of the PE3/TIOC0D/SCK0 pin. |
| 12 | PE3MD0 | 0 | R/W | 000: PE3 I/O (port)<br>001: TIOC0D I/O (MTU2)<br>110: SCK0 I/O (SCI)<br>Other than above: Setting prohibited |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PE2MD2 | 0 | R/W | PE2 Mode |
| 9 | PE2MD1 | 0 | R/W | Select the function of the PE2/TIOC0C/TXD0 pin. |
| 8 | PE2MD0 | 0 | R/W | 000: PE2 I/O (port)<br>001: TIOC0C I/O (MTU2)<br>110: TXD0 output (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PE1MD2 | 0 | R/W | PE1 Mode |
| 5 | PE1MD1 | 0 | R/W | Select the function of the PE1/TIOC0B/RXD0 pin. |
| 4 | PE1MD0 | 0 | R/W | 000: PE1 I/O (port)<br>001: TIOC0B I/O (MTU2)<br>110: RXD0 input (SCI)<br>Other than above: Setting prohibited |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | PE0MD1 | 0 | R/W | PE0 Mode |
| 0 | PE0MD0 | 0 | R/W | Select the function of the PE0/TIOC0A pin.<br>00: PE0 I/O (port)<br>01: TIOC0A I/O (MTU2)<br>Other than above: Setting prohibited |

**SH7132/SH7137:**

- Port E Control Register H2 (PECRH2)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | PE21 MD1 | PE21 MD0 | - | - | PE20 MD1 | PE20 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5 | PE21MD1 | 0 | R/W | PE21 Mode |
| 4 | PE21MD0 | 0 | R/W | Select the function of the PE21/$\overline{\text{WRL}}$/TIOC4DS/$\overline{\text{TRST}}$ pin. Fixed to $\overline{\text{TRST}}$ input when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE21 I/O (port)<br>01: TIOC4DS I/O (MTU2S)<br>10: $\overline{\text{WRL}}$ output (BSC)*<br>Other than above: Setting prohibited |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | PE20MD1 | 0 | R/W | PE20 Mode |
| 0 | PE20MD0 | 0 | R/W | Select the function of the PE20/TIOC4CS/TMS pin. Fixed to TMS input when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE20 I/O (port)<br>01: TIOC4CS I/O (MTU2S)<br>Other than above: Setting prohibited |

Note: * This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

- Port E Control Register H1 (PECRH1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | PE19 MD1 | PE19 MD0 | - | - | PE18 MD1 | PE18 MD0 | - | - | PE17 MD1 | PE17 MD0 | - | PE16 MD2 | PE16 MD1 | PE16 MD0 |
| 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/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15, 14 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 13 | PE19MD1 | 0 | R/W | PE19 Mode |
| 12 | PE19MD0 | 0 | R/W | Select the function of the PE19/$\overline{\text{RD}}$/TIOC4BS/TDO pin. Fixed to TDO output when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE19 I/O (port)<br>01: TIOC4BS I/O (MTU2S)<br>10: $\overline{\text{RD}}$ output (BSC)*<br>Other than above: Setting prohibited |
| 11, 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 9 | PE18MD1 | 0 | R/W | PE18 Mode |
| 8 | PE18MD0 | 0 | R/W | Select the function of the PE18/$\overline{\text{CS1}}$/TIOC4AS/TDI pin. Fixed to TDI input when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE18 I/O (port)<br>01: TIOC4AS I/O (MTU2S)<br>10: $\overline{\text{CS1}}$ output (BSC)*<br>Other than above: Setting prohibited |
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | PE17MD1 | 0 | R/W | PE17 Mode |
| 4 | PE17MD0 | 0 | R/W | Select the function of the PE17/$\overline{\text{CS0}}$/TIOC3DS/TCK pin. Fixed to TCK input when using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>00: PE17 I/O (port)<br>01: TIOC3DS I/O (MTU2S)<br>10: $\overline{\text{CS0}}$ output (BSC)*<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PE16MD2 | 0 | R/W | PE16 Mode |
| 1 | PE16MD1 | 0 | R/W | Select the function of the PE16/$\overline{\text{WAIT}}$/TIOC3BS/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$ pin. Fixed to $\overline{\text{ASEBRKAK}}$ output/$\overline{\text{ASEBRK}}$ input when using the E10A ($\overline{\text{ASEMD0}}$ = low). |
| 0 | PE16MD0 | 0 | R/W | 000: PE16 I/O (port)<br>001: TIOC3BS I/O (MTU2S)<br>010: $\overline{\text{WAIT}}$ input (BSC)*<br>Other than above: Setting prohibited |

Note: * This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

- Port E Control Register L4 (PECRL4)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE15 MD2 | PE15 MD1 | PE15 MD0 | - | PE14 MD2 | PE14 MD1 | PE14 MD0 | - | - | PE13 MD1 | PE13 MD0 | - | PE12 MD2 | PE12 MD1 | PE12 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R | R/W | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 14 | PE15MD2 | 0 | R/W | PE15 Mode |
| 13 | PE15MD1 | 0 | R/W | Select the function of the PE15/TIOC4D/$\overline{\text{IRQOUT}}$ pin. |
| 12 | PE15MD0 | 0 | R/W | 000: PE15 I/O (port)<br>001: TIOC4D I/O (MTU2)<br>011: $\overline{\text{IRQOUT}}$ output (INTC)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PE14MD2 | 0 | R/W | PE14 Mode |
| 9 | PE14MD1 | 0 | R/W | Select the function of the PE14/TIOC4C pin. |
| 8 | PE14MD0 | 0 | R/W | 000: PE14 I/O (port)<br>001: TIOC4C I/O (MTU2)<br>Other than above: Setting prohibited |
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5 | PE13MD1 | 0 | R/W | PE13 Mode |
| 4 | PE13MD0 | 0 | R/W | Select the function of the PE13/TIOC4B/$\overline{\text{MRES}}$ pin.<br>00: PE13 I/O (port)<br>01: TIOC4B I/O (MTU2)<br>10: $\overline{\text{MRES}}$ input (INTC)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PE12MD2 | 0 | R/W | PE12 Mode |
| 1 | PE12MD1 | 0 | R/W | Select the function of the PE12/TIOC4A pin. |
| 0 | PE12MD0 | 0 | R/W | 000: PE12 I/O (port)<br>001: TIOC4A I/O (MTU2)<br>Other than above: Setting prohibited |

- Port E Control Register L3 (PECRL3)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE11 MD2 | PE11 MD1 | PE11 MD0 | - | PE10 MD2 | PE10 MD1 | PE10 MD0 | - | PE9 MD2 | PE9 MD1 | PE9 MD0 | - | PE8 MD2 | PE8 MD1 | PE8 MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0*1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0*1 | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Note: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PE11MD2 | 0 | R/W | PE11 Mode |
| 13 | PE11MD1 | 0 | R/W | Select the function of the PE11/TIOC3D pin. |
| 12 | PE11MD0 | 0 | R/W | 000: PE11 I/O (port)<br>001: TIOC3D I/O (MTU2)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PE10MD2 | 0*1 | R/W | PE10 Mode |
| 9 | PE10MD1 | 0 | R/W | Select the function of the PE10/$\overline{\text{CS0}}$/TIOC3C pin. |
| 8 | PE10MD0 | 0 | R/W | 000: PE10 I/O (port)<br>001: TIOC3C I/O (MTU2)<br>100: $\overline{\text{CS0}}$ output (BSC)*2<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PE9MD2 | 0 | R/W | PE9 Mode |
| 5 | PE9MD1 | 0 | R/W | Select the function of the PE9/TIOC3B pin. |
| 4 | PE9MD0 | 0 | R/W | 000: PE9 I/O (port)<br>001: TIOC3B I/O (MTU2)<br>Other than above: Setting prohibited |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PE8MD2 | 0*[1] | R/W | PE8 Mode |
| 1 | PE8MD1 | 0 | R/W | Select the function of the PE8/A15/TIOC3A pin. |
| 0 | PE8MD0 | 0 | R/W | 000: PE8 I/O (port)<br>001: TIOC3A I/O (MTU2)<br>100: A15 output (BSC)*[2]<br>Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

- Port E Control Register L2 (PECRL2)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE7 MD2 | PE7 MD1 | PE7 MD0 | - | PE6 MD2 | PE6 MD1 | PE6 MD0 | - | PE5 MD2 | PE5 MD1 | PE5 MD0 | - | PE4 MD2 | PE4 MD1 | PE4 MD0 |
| Initial value: | 0 | 0*[1] | 0 | 0 | 0 | 0*[1] | 0 | 0 | 0 | 0*[1] | 0 | 0 | 0 | 0*[1] | 0 | 0 |
| R/W: | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Note: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PE7MD2 | 0*[1] | R/W | PE7 Mode |
| 13 | PE7MD1 | 0 | R/W | Select the function of the PE7/A14/TIOC2B pin. |
| 12 | PE7MD0 | 0 | R/W | 000: PE7 I/O (port)<br>001: TIOC2B I/O (MTU2)<br>100: A14 output (BSC)*[2]<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 | PE6MD2 | 0*[1] | R/W | PE6 Mode |
| 9 | PE6MD1 | 0 | R/W | Select the function of the PE6/A13/TIOC2A/SCK1 pin. |
| 8 | PE6MD0 | 0 | R/W | 000: PE6 I/O (port)<br>001: TIOC2A I/O (MTU2)<br>100: A13 output (BSC)*[2]<br>110: SCK1 I/O (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | PE5MD2 | 0*[1] | R/W | PE5 Mode |
| 5 | PE5MD1 | 0 | R/W | Select the function of the PE5/A12/TIOC1B/TXD1 pin. |
| 4 | PE5MD0 | 0 | R/W | 000: PE5 I/O (port)<br>001: TIOC1B I/O (MTU2)<br>100: A12 output (BSC)*[2]<br>110: TXD1 output (SCI)<br>Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | PE4MD2 | 0*[1] | R/W | PE4 Mode |
| 1 | PE4MD1 | 0 | R/W | Select the function of the PE4/A11/TIOC1A/RXD1 pin. |
| 0 | PE4MD0 | 0 | R/W | 000: PE4 I/O (port)<br>001: TIOC1A I/O (MTU2)<br>100: A11 output (BSC)*[2]<br>110: RXD1 input (SCI)<br>Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

- Port E Control Register L1 (PECRL1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE3 MD2 | PE3 MD1 | PE3 MD0 | - | PE2 MD2 | PE2 MD1 | PE2 MD0 | - | PE1 MD2 | PE1 MD1 | PE1 MD0 | - | - | PE0 MD1 | PE0 MD0 |
| 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<br>This bit is always read as 0. The write value should always be 0. |
| 14 | PE3MD2 | 0 | R/W | PE3 Mode |
| 13 | PE3MD1 | 0 | R/W | Select the function of the PE3/TIOC0D/SCK0 pin. |
| 12 | PE3MD0 | 0 | R/W | 000: PE3 I/O (port)<br>001: TIOC0D I/O (MTU2)<br>110: SCK0 I/O (SCI)<br>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 | PE2MD2 | 0 | R/W | PE2 Mode |
| 9 | PE2MD1 | 0 | R/W | Select the function of the PE2/TIOC0C/TXD0 pin. |
| 8 | PE2MD0 | 0 | R/W | 000: PE2 I/O (port)<br>001: TIOC0C I/O (MTU2)<br>110: TXD0 output (SCI)<br>Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | PE1MD2 | 0 | R/W | PE1 Mode |
| 5 | PE1MD1 | 0 | R/W | Select the function of the PE1/TIOC0B/RXD0 pin. |
| 4 | PE1MD0 | 0 | R/W | 000: PE1 I/O (port)<br>001: TIOC0B I/O (MTU2)<br>110: RXD0 input (SCI)<br>Other than above: Setting prohibited |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | PE0MD1 | 0 | R/W | PE0 Mode |
| 0 | PE0MD0 | 0 | R/W | Select the function of the PE0/TIOC0A pin.<br>00: PE0 I/O (port)<br>01: TIOC0A I/O (MTU2)<br>Other than above: Setting prohibited |

### 20.1.9 IRQOUT Function Control Register (IFCR)

IFCR is a 16-bit readable/writable register that is used to control the IRQOUT pin output when it is selected as the multiplexed pin function by port E control register L4 (PECRL4). When PECRL4 selects another function, the IFCR setting does not affect the pin function.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | - | - | - | - | IRQ MD1 | IRQ MD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | IRQMD1 | 0 | R/W | Port E $\overline{\text{IRQOUT}}$ Pin Function Select |
| 0 | IRQMD0 | 0 | R/W | Select the $\overline{\text{IRQOUT}}$ pin function when bits 14 to 12 (PE15MD2 to PE15MD0) in PECRL4 are set to B'011.<br>00: Interrupt request accept signal output<br>01: Setting prohibited<br>10: Interrupt request accept signal output<br>11: Always high-level output |

## 20.2 Usage Notes

1. In this LSI, the same function is available as a multiplexed function on multiple pins. This approach is intended to increase the number of selectable pin functions and to allow the easier design of boards. If two or more pins are specified for one function, however, there are two cautions shown below.

   — When the pin function is input

   Signals input to several pins are formed as one signal through OR or AND logic and the signal is transmitted into the LSI. Therefore, a signal that differs from the input signals may be transmitted to the LSI depending on the input signals in other pins that have the same functions. Table 20.14 shows the transmit forms of input functions allocated to several pins. When using one of the functions shown below in multiple pins, use it with care of signal polarity considering the transmit forms.

**Table 20.14 Transmit Forms of Input Functions Allocated to Multiple Pins**

| OR Type | AND Type |
|---|---|
| SCK0 to SCK2, RXD0 to RXD2 | IRQ0 to IRQ3, $\overline{\text{WAIT}}$, $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE5}}$ |

   OR type: Signals input to several pins are formed as one signal through OR logic and the signal is transmitted into the LSI.

   AND type: Signals input to several pins are formed as one signal through AND logic and the signal is transmitted into the LSI.

   — When the pin function is output

   Each selected pin can output the same function.

2. When the port input is switched from a low level to the IRQ edge for the pins that are multiplexed with input/output and IRQ, the corresponding edge is detected.
3. Do not set functions other than those specified in tables 20.10 to 20.12. Otherwise, correct operation cannot be guaranteed.
4. PFC setting in single-chip mode (MCU operating mode 3)

   In single-chip mode, do not set the PFC to select address bus, data bus, bus control, or the $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, or CK signals. If they are selected, address bus signals function as high- or low-level outputs, data bus signals function as high-impedance outputs, and the other output signals function as high-level outputs. As $\overline{\text{BREQ}}$ and $\overline{\text{WAIT}}$ function as inputs, do not leave them open. However, the bus-mastership-request inputs and external waits are disabled.

# Section 21 I/O Ports

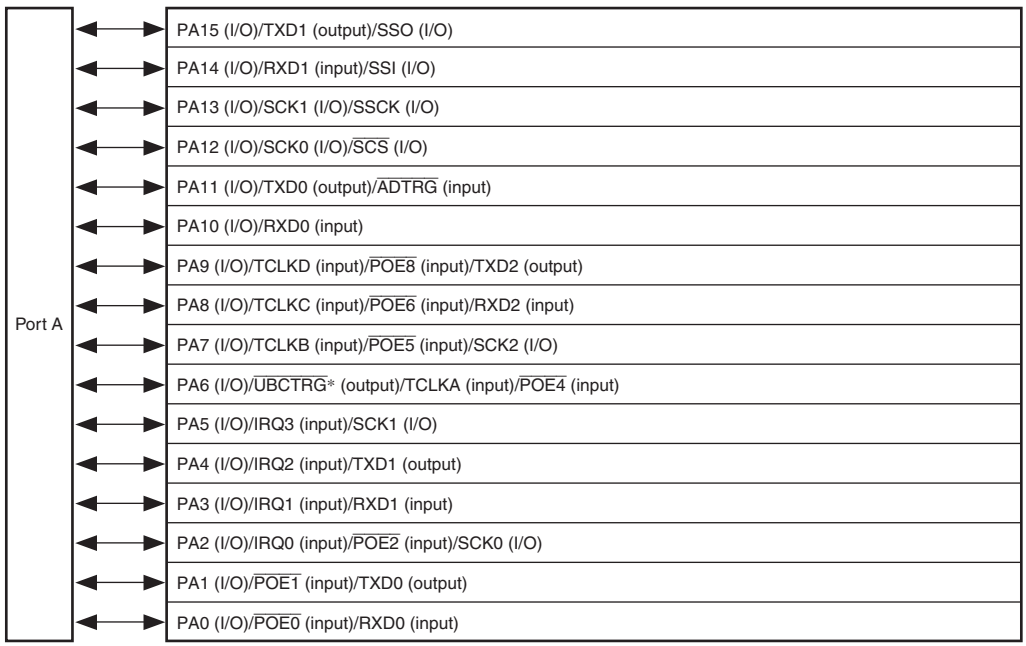
The SH7131 and SH7136 have four ports: A, B, E, and F. Port A is a 16-bit I/O port, port B is a 6-bit I/O port, and port E is a 22-bit I/O port. Port F is a 12-bit input-only port.

The SH7132 and SH7137 have five ports: A, B, D, E, and F. Port A is a 16-bit I/O port, port B is an 8-bit I/O port, port D is an 11-bit I/O port, and port E is a 22-bit I/O port. Port F is a 16-bit input-only port.

All the port pins are multiplexed as general input/output pins and special function pins. The functions of the multiplex pins are selected by means of the pin function controller (PFC). Each port is provided with a data register for storing the pin data.

## 21.1 Port A

Port A in the SH7131 and SH7136 is an input/output port with the 16 pins shown in figure 21.1.




Note: * SH7136 only.

**Figure 21.1 Port A (SH7131/SH7136)**

Port A in the SH7132 and SH7137 is an input/output port with the 16 pins shown in figure 21.2.

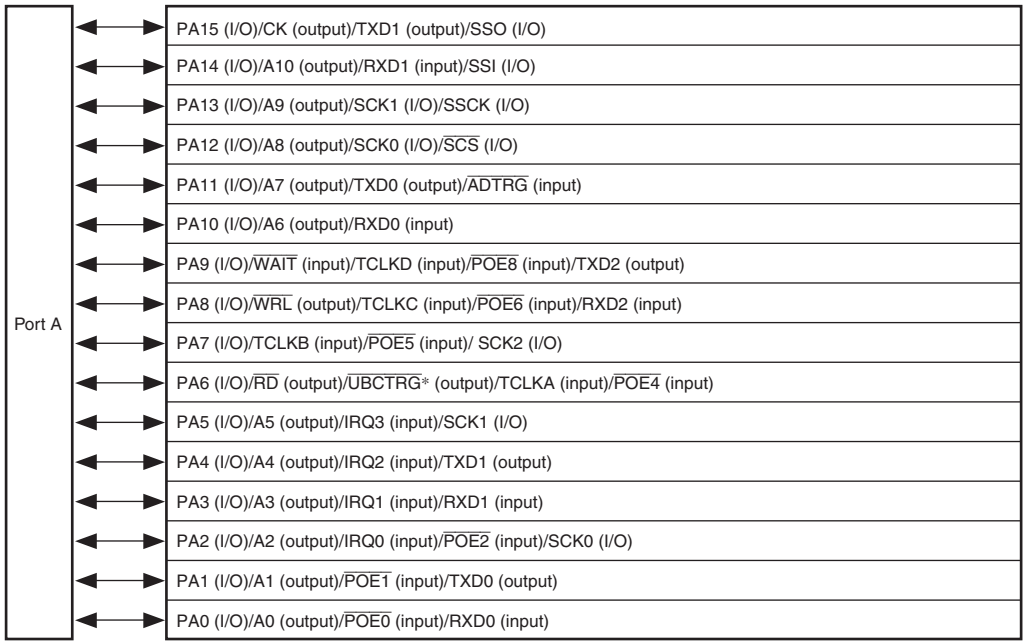



**Figure 21.2 Port A (SH7132/SH7137)**

### 21.1.1 Register Descriptions

Port A is a 16-bit input/output port. Port A has the following registers. For details on register addresses and register states during each processing, refer to section 25, List of Registers.

**Table 21.1 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Port A data register L | PADRL | R/W | H'0000 | H'FFFFD102 | 8, 16 |
| Port A port register L | PAPRL | R | H'xxxx | H'FFFFD11E | 8, 16 |

### 21.1.2 Port A Data Register L (PADRL)

The port A data register L (PADRL) is a 16-bit readable/writable register that stores port A data. Bits PA15DR to PA0DR correspond to pins PA15 to PA0 (multiplexed functions omitted here).

When a pin function is general output, if a value is written to PADRL, that value is output directly from the pin, and if PADRL is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PADRL is read, the pin state, not the register value, is returned directly. If a value is written to PADRL, although that value is written into PADRL, it does not affect the pin state. Table 21.2 summarizes port A data register read/write operations.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PA15 DR | PA14 DR | PA13 DR | PA12 DR | PA11 DR | PA10 DR | PA9 DR | PA8 DR | PA7 DR | PA6 DR | PA5 DR | PA4 DR | PA3 DR | PA2 DR | PA1 DR | PA0 DR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | PA15DR | 0 | R/W | See table 21.2. |
| 14 | PA14DR | 0 | R/W | |
| 13 | PA13DR | 0 | R/W | |
| 12 | PA12DR | 0 | R/W | |
| 11 | PA11DR | 0 | R/W | |
| 10 | PA10DR | 0 | R/W | |
| 9 | PA9DR | 0 | R/W | |
| 8 | PA8DR | 0 | R/W | |
| 7 | PA7DR | 0 | R/W | |
| 6 | PA6DR | 0 | R/W | |
| 5 | PA5DR | 0 | R/W | |
| 4 | PA4DR | 0 | R/W | |
| 3 | PA3DR | 0 | R/W | |
| 2 | PA2DR | 0 | R/W | |
| 1 | PA1DR | 0 | R/W | |
| 0 | PA0DR | 0 | R/W | |

**Table 21.2 Port A Data Register L (PADRL) Read/Write Operations**

- PADRL Bits 15 to 0

| PAIOR | Pin Function | Read | Write |
|---|---|---|---|
| 0 | General input | Pin state | Can write to PADRL, but it has no effect on pin state |
| | Other than general input | Pin state | Can write to PADRL, but it has no effect on pin state |
| 1 | General output | PADRL value | Value written is output from pin |
| | Other than general output | PADRL value | Can write to PADRL, but it has no effect on pin state |

### 21.1.3 Port A Port Register L (PAPRL)

The port A port register L (PAPRL) is a 16-bit read-only register that always returns the states of the pins regardless of the PFC setting. Bits PA15PR to PA0PR correspond to pins PA15 to PA0 (multiplexed functions omitted here).

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PA15 PR | PA14 PR | PA13 PR | PA12 PR | PA11 PR | PA10 PR | PA9 PR | PA8 PR | PA7 PR | PA6 PR | PA5 PR | PA4 PR | PA3 PR | PA2 PR | PA1 PR | PA0 PR |
| Initial value: | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | PA15PR | Pin state | R | The pin state is returned regardless of the PFC setting. These bits cannot be modified. |
| 14 | PA14PR | Pin state | R | |
| 13 | PA13PR | Pin state | R | |
| 12 | PA12PR | Pin state | R | |
| 11 | PA11PR | Pin state | R | |
| 10 | PA10PR | Pin state | R | |
| 9 | PA9PR | Pin state | R | |
| 8 | PA8PR | Pin state | R | |
| 7 | PA7PR | Pin state | R | |
| 6 | PA6PR | Pin state | R | |
| 5 | PA5PR | Pin state | R | |
| 4 | PA4PR | Pin state | R | |
| 3 | PA3PR | Pin state | R | |
| 2 | PA2PR | Pin state | R | |
| 1 | PA1PR | Pin state | R | |
| 0 | PA0PR | Pin state | R | |

## 21.2 Port B

Port B in the SH7131 and SH7136 is an input/output port with the six pins shown in figure 21.3.

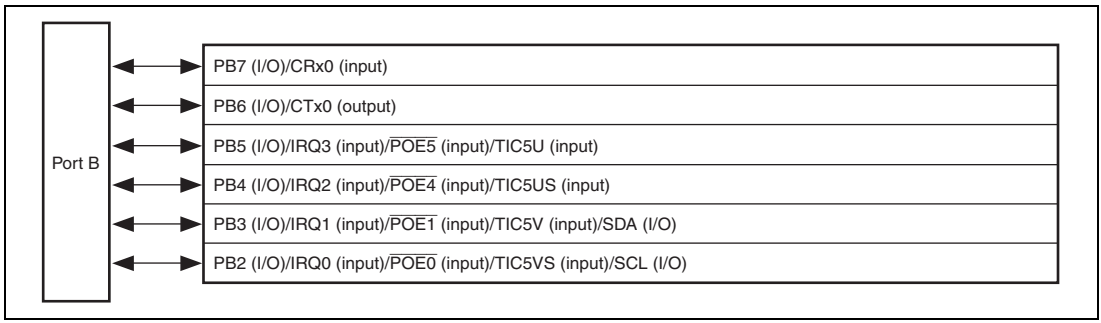



**Figure 21.3 Port B (SH7131/SH7136)**

Port B in the SH7132 and SH7137 is an input/output port with the eight pins shown in figure 21.4.

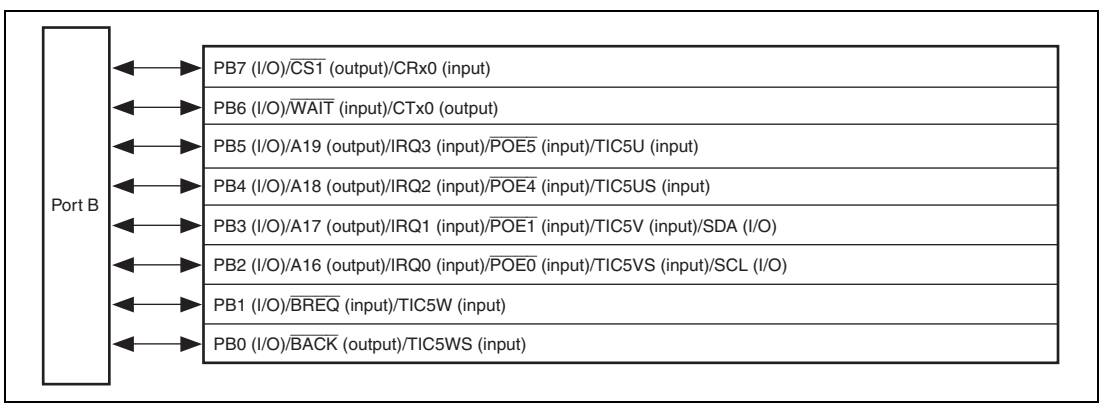



**Figure 21.4 Port B (SH7132/SH7137)**

### 21.2.1 Register Descriptions

Port B is a 6-bit input/output port in the SH7131 and SH7136, an 8-bit input/output port in the SH7132 and SH7137. Port B has the following registers. For details on register addresses and register states during each processing, refer to section 25, List of Registers.

**Table 21.3 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Port B data register L | PBDRL | R/W | H'0000 | H'FFFFD182 | 8, 16 |
| Port B port register L | PBPRL | R | H'00xx | H'FFFFD19E | 8, 16 |

### 21.2.2 Port B Data Register L (PBDRL)

The port B data register L (PBDRL) is a 16-bit readable/writable register that stores port B data. Bits PB7DR to PB2DR correspond to pins PB7 to PB2, respectively (multiplexed functions omitted here) in the SH7131 and SH7136. Bits PB7DR to PB0DR correspond to pins PB7 to PB0, respectively (multiplexed functions omitted here) in the SH7132 and SH7137.

When a pin function is general output, if a value is written to PBDRL, that value is output directly from the pin, and if PBDRL is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PBDRL is read, the pin state, not the register value, is returned directly. If a value is written to PBDRL, although that value is written into PBDRL, it does not affect the pin state. Table 21.4 summarizes port B data register read/write operations.

- PBDRL (SH7131/SH7136)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | PB10 DR | PB9 DR | PB8 DR | PB7 DR | PB6 DR | PB5 DR | PB4 DR | PB3 DR | PB2 DR | - | - |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 7 | PB7DR | 0 | R/W | See table 21.4. |
| 6 | PB6DR | 0 | R/W | |
| 5 | PB5DR | 0 | R/W | |
| 4 | PB4DR | 0 | R/W | |
| 3 | PB3DR | 0 | R/W | |
| 2 | PB2DR | 0 | R/W | |
| 1, 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

- PBDRL (SH7132/SH7137)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | PB7 DR | PB7 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 | 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<br>These bits are always read as 0. The write value should always be 0. |
| 7 | PB7DR | 0 | R/W | See table 21.4. |
| 6 | PB6DR | 0 | R/W | |
| 5 | PB5DR | 0 | R/W | |
| 4 | PB4DR | 0 | R/W | |
| 3 | PB3DR | 0 | R/W | |
| 2 | PB2DR | 0 | R/W | |
| 1 | PB1DR | 0 | R/W | |
| 0 | PB0DR | 0 | R/W | |

**Table 21.4 Port B Data Register (PBDR) Read/Write Operations**

- PBDRL Bits 7 to 0

| PBIOR | Pin Function | Read | Write |
|---|---|---|---|
| 0 | General input | Pin state | Can write to PBDRL, but it has no effect on pin state |
| | Other than general input | Pin state | Can write to PBDRL, but it has no effect on pin state |
| 1 | General output | PBDRL value | Value written is output from pin |
| | Other than general output | PBDRL value | Can write to PBDRL, but it has no effect on pin state |

### 21.2.3 Port B Port Register L (PBPRL)

The port B port register L (PBPRL) is a 16-bit read-only register that always returns the states of the pins regardless of the PFC setting. Bits PB7PR to PB2PR correspond to pins PB7 to PB2, respectively (multiplexed functions omitted here) in the SH7131 and SH7136. Bits PB7PR to PB0PR correspond to pins PB7 to PB0, respectively (multiplexed functions omitted here) in the SH7132 and SH7137.

- PBPRL (SH7131/SH7136)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | PB7PR | PB6PR | PB5PR | PB4PR | PB3PR | PB2PR | - | - |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | * | * | * | * | * | * | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 7 | PB7PR | Pin state | R | The pin state is returned regardless of the PFC setting. These bits cannot be modified. |
| 6 | PB6PR | Pin state | R | |
| 5 | PB5PR | Pin state | R | |
| 4 | PB4PR | Pin state | R | |
| 3 | PB3PR | Pin state | R | |
| 2 | PB2PR | Pin state | R | |
| 1, 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

- PBPRL (SH7132/SH7137)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | PB7PR | PB6PR | PB5PR | PB4PR | PB3PR | PB2PR | PB1PR | PB0PR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | * | * | * | * | * | * | * | * |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 7 | PB7PR | Pin state | R | The pin state is returned regardless of the PFC setting. These bits cannot be modified. |
| 6 | PB6PR | Pin state | R | |
| 5 | PB5PR | Pin state | R | |
| 4 | PB4PR | Pin state | R | |
| 3 | PB3PR | Pin state | R | |
| 2 | PB2PR | Pin state | R | |
| 1 | PB1PR | Pin state | R | |
| 0 | PB0PR | Pin state | R | |

## 21.3 Port D (SH7132/SH7137 Only)

Port D in the SH7132 and SH7137 is an input/output port with the 11 pins shown in figure 21.5.

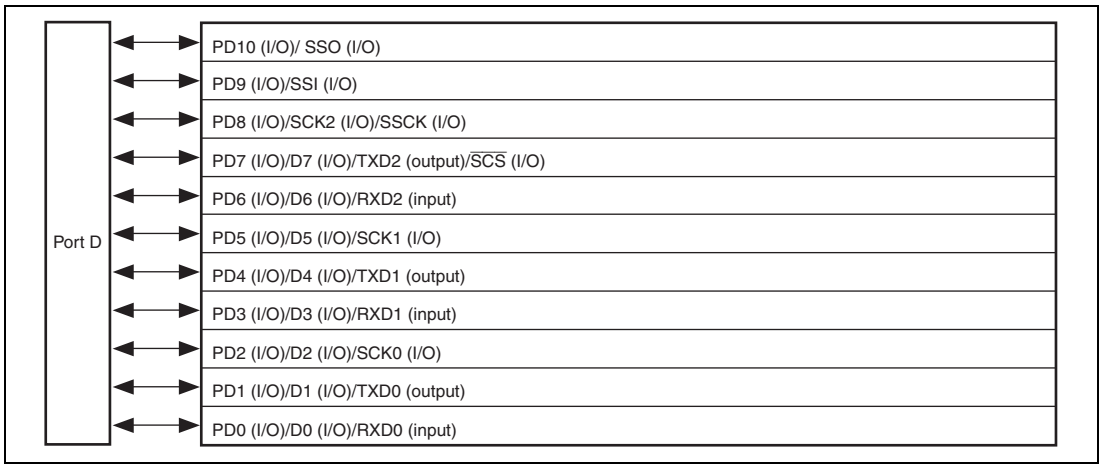



**Figure 21.5 Port D**

### 21.3.1 Register Descriptions

Port D is an 11-bit input/output port. Note that port D is not available in the SH7131 and SH7136. Port D has the following registers. For details on register addresses and register states during each processing, refer to section 25, List of Registers.

**Table 21.5 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Port D data register L | PDDRL | R/W | H'0000 | H'FFFFD282 | 8, 16 |
| Port D port register L | PDPRL | R | H'xxxx | H'FFFFD29E | 8, 16 |

### 21.3.2 Port D Data Register L (PDDRL)

The port D data register L (PDDRL) is a 16-bit readable/writable register that stores port D data. Bits PD10DR to PD0DR correspond to pins PD10 to PD0 (multiplexed functions omitted here).

When a pin function is general output, if a value is written to PDDRL, that value is output directly from the pin, and if PDDRL is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PDDRL is read, the pin state, not the register value, is returned directly. If a value is written to PDDRL, although that value is written into PDDRL, it does not affect the pin state. Table 21.6 summarizes port D data register read/write operations.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | PB7 DR | PB7 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 | 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 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 10 | PD10DR | 0 | R/W | See table 21.6. |
| 9 | PD9DR | 0 | R/W | |
| 8 | PD8DR | 0 | R/W | |
| 7 | PD7DR | 0 | R/W | |
| 6 | PD6DR | 0 | R/W | |
| 5 | PD5DR | 0 | R/W | |
| 4 | PD4DR | 0 | R/W | |
| 3 | PD3DR | 0 | R/W | |
| 2 | PD2DR | 0 | R/W | |
| 1 | PD1DR | 0 | R/W | |
| 0 | PD0DR | 0 | R/W | |

**Table 21.6 Port D Data Register L (PDDRL) Read/Write Operations**

- PDDRL Bits 10 to 0

| PDIOR | Pin Function | Read | Write |
|---|---|---|---|
| 0 | General input | Pin state | Can write to PDDRL, but it has no effect on pin state |
| | Other than general input | Pin state | Can write to PDDRL, but it has no effect on pin state |
| 1 | General output | PDDRL value | Value written is output from pin |
| | Other than general output | PDDRL value | Can write to PDDRL, but it has no effect on pin state |

### 21.3.3 Port D Port Register L (PDPRL)

The port D port register L (PDPRL) is a 16-bit read-only register that always returns the states of the pins regardless of the PFC setting. Bits PD10PR to PD0PR correspond to pins PD10 to PD0 (multiplexed functions omitted here).

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | PD10 PR | PD9 PR | PD8 PR | PD7 PR | PD6 PR | PD5 PR | PD4 PR | PD3 PR | PD2 PR | PD1 PR | PD0 PR |
| Initial value: | 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 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 10 | PD10PR | Pin state | R | The pin state is returned regardless of the PFC setting. These bits cannot be modified. |
| 9 | PD9PR | Pin state | R | |
| 8 | PD8PR | Pin state | R | |
| 7 | PD7PR | Pin state | R | |
| 6 | PD6PR | Pin state | R | |
| 5 | PD5PR | Pin state | R | |
| 4 | PD4PR | Pin state | R | |
| 3 | PD3PR | Pin state | R | |
| 2 | PD2PR | Pin state | R | |
| 1 | PD1PR | Pin state | R | |
| 0 | PD0PR | Pin state | R | |

## 21.4 Port E

Port E in the SH7131 and SH7136 is an input/output port with the 22 pins shown in figure 21.6.

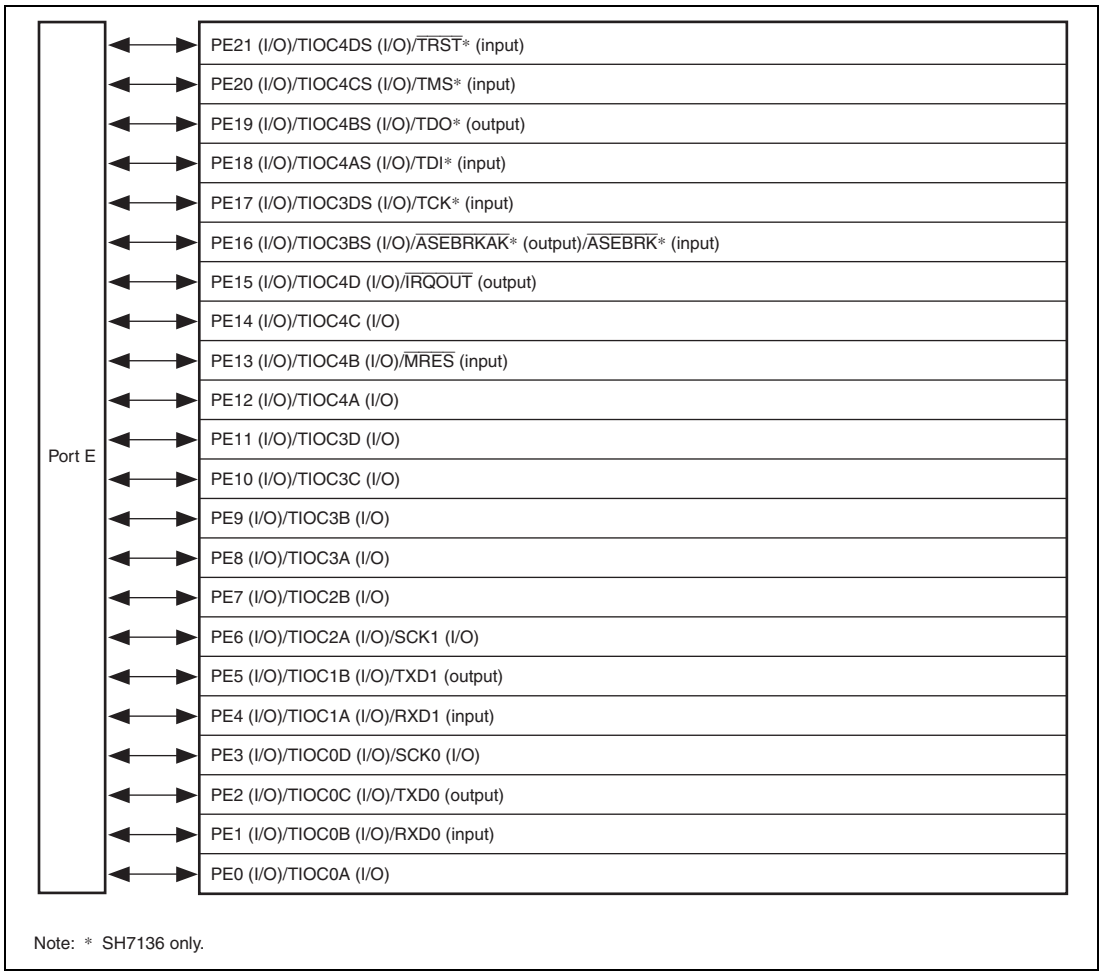



**Figure 21.6 Port E (SH7131/SH7136)**

Port E in the SH7132 and SH7137 is an input/output port with the 22 pins shown in figure 21.7.

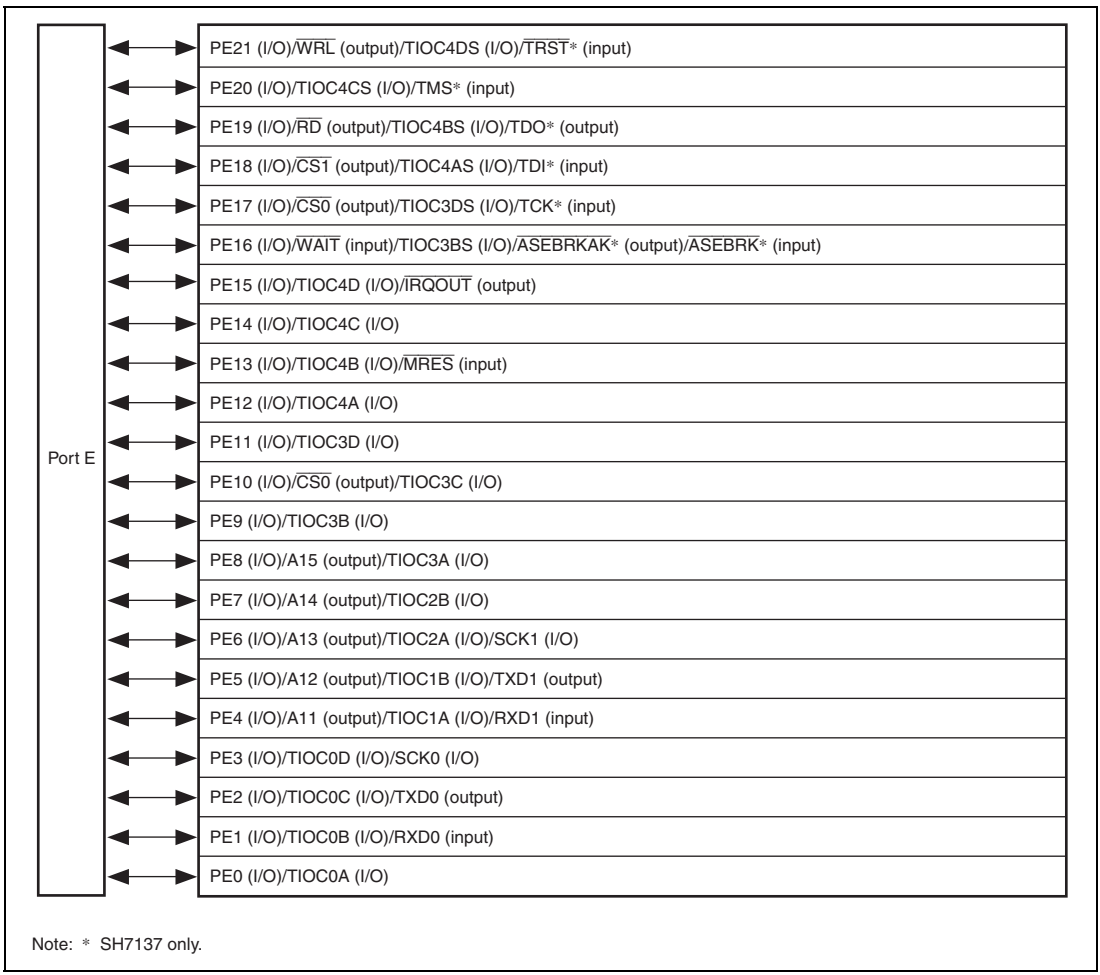



**Figure 21.7 Port E (SH7132/SH7137)**

### 21.4.1 Register Descriptions

Port E is a 22-bit input/output port. Port E has the following registers. For details on register addresses and register states during each processing, refer to section 25, List of Registers.

**Table 21.7 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Port E data register H | PEDRH | R/W | H'0000 | H'FFFFD300 | 8, 16, 32 |
| Port E data register L | PEDRL | R/W | H'0000 | H'FFFFD302 | 8, 16 |
| Port E port register H | PEPRH | R | H'00xx | H'FFFFD31C | 8, 16, 32 |
| Port E port register L | PEPRL | R | H'xxxx | H'FFFFD31E | 8, 16 |

### 21.4.2 Port E Data Registers H and L (PEDRH and PEDRL)

The port E data registers H and L (PEDRH and PEDRL) are 16-bit readable/writable registers that store port E data. Bits PE21DR to PE0DR correspond to pins PE21 to PE0, respectively (multiplexed functions omitted here).

When a pin function is general output, if a value is written to PEDRH or PEDRL, that value is output directly from the pin, and if PEDRH or PEDRL is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PEDRH or PEDRL is read, the pin state, not the register value, is returned directly. If a value is written to PEDRH or PEDRL, although that value is written into PEDRH or PEDRL, it does not affect the pin state. Table 21.8 summarizes port E data register read/write operations.

- PEDRH

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | PE21 DR | PE20 DR | PE19 DR | PE18 DR | PE17 DR | PE16 DR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5 | PE21DR | 0 | R/W | See table 21.8. |
| 4 | PE20DR | 0 | R/W | |
| 3 | PE19DR | 0 | R/W | |
| 2 | PE18DR | 0 | R/W | |
| 1 | PE17DR | 0 | R/W | |
| 0 | PE16DR | 0 | R/W | |

- PEDRL

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | PE15DR | 0 | R/W | See table 21.8. |
| 14 | PE14DR | 0 | R/W | |
| 13 | PE13DR | 0 | R/W | |
| 12 | PE12DR | 0 | R/W | |
| 11 | PE11DR | 0 | R/W | |
| 10 | PE10DR | 0 | R/W | |
| 9 | PE9DR | 0 | R/W | |
| 8 | PE8DR | 0 | R/W | |
| 7 | PE7DR | 0 | R/W | |
| 6 | PE6DR | 0 | R/W | |
| 5 | PE5DR | 0 | R/W | |
| 4 | PE4DR | 0 | R/W | |
| 3 | PE3DR | 0 | R/W | |
| 2 | PE2DR | 0 | R/W | |
| 1 | PE1DR | 0 | R/W | |
| 0 | PE0DR | 0 | R/W | |

**Table 21.8 Port E Data Register (PEDR) Read/Write Operations**

- PEDRH Bits 5 to 0 and PEDRL Bits 15 to 0

| PEIOR | Pin Function | Read | Write |
|---|---|---|---|
| 0 | General input | Pin state | Can write to PEDRH and PEDRL, but it has no effect on pin state |
| | Other than general input | Pin state | Can write to PEDRH and PEDRL, but it has no effect on pin state |
| 1 | General output | PEDRH or PEDRL value | Value written is output from pin |
| | Other than general output | PEDRH or PEDRL value | Can write to PEDRH and PEDRL, but it has no effect on pin state |

### 21.4.3 Port E Port Registers H and L (PEPRH and PEPRL)

The port E port registers H and L (PEPRH and PEPRL) are 16-bit read-only registers that always return the states of the pins regardless of the PFC setting. Bits PE21PR to PE0PR correspond to pins PE21 to PE0, respectively (multiplexed functions omitted here).

- PEPRH

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | PE21 PR | PE20 PR | PE19 PR | PE18 PR | PE17 PR | PE16 PR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | * | * | * | * | * | * |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5 | PE21PR | Pin state | R | The pin state is returned regardless of the PFC setting. These bits cannot be modified. |
| 4 | PE20PR | Pin state | R | |
| 3 | PE19PR | Pin state | R | |
| 2 | PE18PR | Pin state | R | |
| 1 | PE17PR | Pin state | R | |
| 0 | PE16PR | Pin state | R | |

- PEPRL

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PE15 PR | PE14 PR | PE13 PR | PE12 PR | PE11 PR | PE10 PR | PE9 PR | PE8 PR | PE7 PR | PE6 PR | PE5 PR | PE4 PR | PE3 PR | PE2 PR | PE1 PR | PE0 PR |
| Initial value: | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | PE15PR | Pin state | R | The pin state is returned regardless of the PFC setting. These bits cannot be modified. |
| 14 | PE14PR | Pin state | R | |
| 13 | PE13PR | Pin state | R | |
| 12 | PE12PR | Pin state | R | |
| 11 | PE11PR | Pin state | R | |
| 10 | PE10PR | Pin state | R | |
| 9 | PE9PR | Pin state | R | |
| 8 | PE8PR | Pin state | R | |
| 7 | PE7PR | Pin state | R | |
| 6 | PE6PR | Pin state | R | |
| 5 | PE5PR | Pin state | R | |
| 4 | PE4PR | Pin state | R | |
| 3 | PE3PR | Pin state | R | |
| 2 | PE2PR | Pin state | R | |
| 1 | PE1PR | Pin state | R | |
| 0 | PE0PR | Pin state | R | |

## 21.5 Port F

Port F in the SH7131 and SH7136 is an input-only port with the 12 pins shown in figure 21.8.

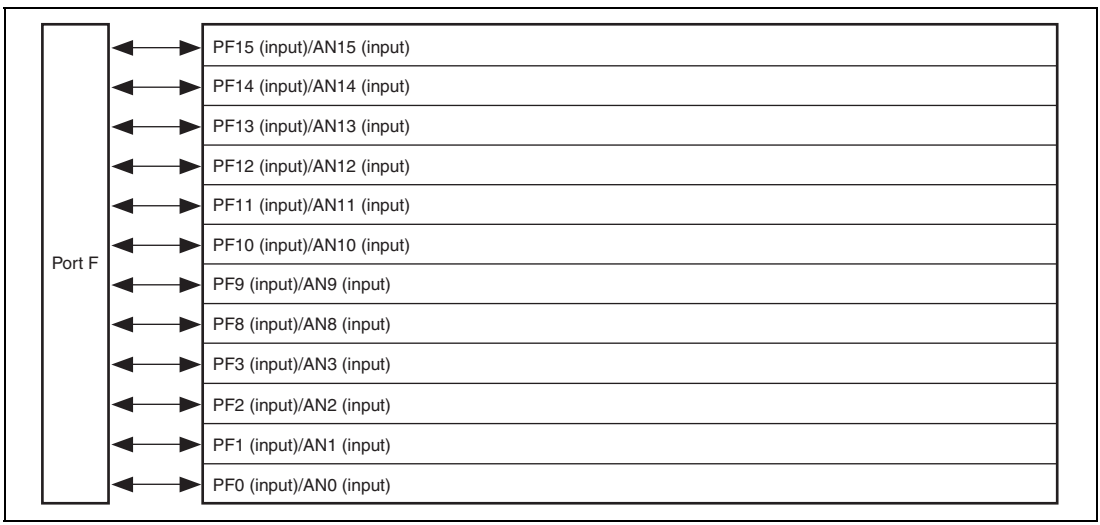



**Figure 21.8 Port F (SH7131/SH7136)**

Port F in the SH7132 and SH7137 is an input-only port with the 16 pins shown in figure 21.9.

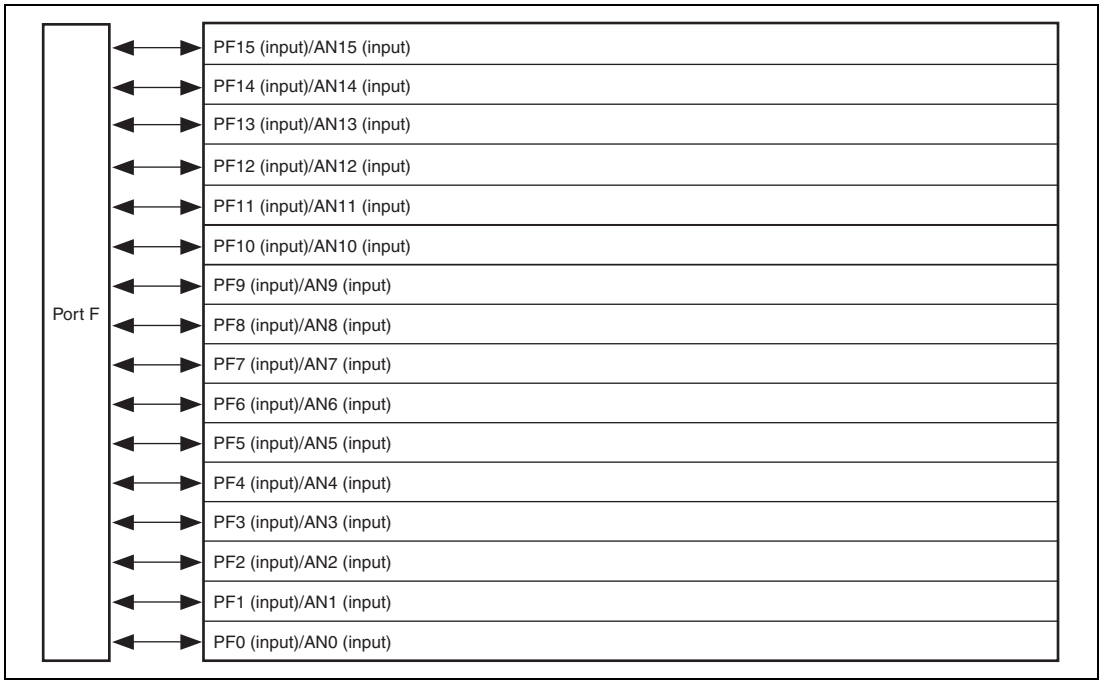



**Figure 21.9 Port F (SH7132/SH7137)**

### 21.5.1 Register Descriptions

Port F is a 12-bit input-only port in the SH7131 and SH7136, and 16-bit input-only port in the SH7132 and SH7137. Port F has the following register. For details on register addresses and register states during each processing, refer to section 25, List of Registers.

**Table 21.9 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Port F data register L | PFDRL | R | H'xxxx | H'FFFFD382 | 8, 16 |

### 21.5.2 Port F Data Register L (PFDRL)

The port F data register L (PFDRL) is a 16-bit read-only register that stores port F data. Bits PF15DR to PF8DR and PF3DR to PF0DR correspond to pins PF15 to PF8 and PF3 to PF0, respectively (multiplexed functions omitted here) in the SH7131 and SH7136. Bits PF15DR to PF0DR correspond to pins PF15 to PF0, respectively (multiplexed functions omitted here) in the SH7132 and SH7137.

Any value written into these bits is ignored, and there is no effect on the state of the pins. When any of the bits are read, the pin state rather than the bit value is read directly. However, when an A/D converter analog input is being sampled, values of 1 are read out. Table 21.10 summarizes port F data register L read/write operations.

- PFDRL (SH7131/SH7136)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PF15 DR | PF14 DR | PF13 DR | PF12 DR | PF11 DR | PF10 DR | PF9 DR | PF8 DR | - | - | - | - | PF3 DR | PF2 DR | PF1 DR | PF0 DR |
| Initial value: | * | * | * | * | * | * | * | * | 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 | PF15DR | Pin state | R | See table 21.10. |
| 14 | PF14DR | Pin state | R | |
| 13 | PF13DR | Pin state | R | |
| 12 | PF12DR | Pin state | R | |
| 11 | PF11DR | Pin state | R | |
| 10 | PF10DR | Pin state | R | |
| 9 | PF9DR | Pin state | R | |
| 8 | PF8DR | Pin state | R | |
| 7 to 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 3 | PF3DR | Pin state | R | See table 21.10. |
| 2 | PF2DR | Pin state | R | |
| 1 | PF1DR | Pin state | R | |
| 0 | PF0DR | Pin state | R | |

- PFDRL (SH7132/SH7137)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PF15 DR | PF14 DR | PF13 DR | PF12 DR | PF11 DR | PF10 DR | PF9 DR | PF8 DR | PF7 DR | PF6 DR | PF5 DR | PF4 DR | PG3 DR | PF2 DR | PF1 DR | PF0 DR |
| Initial value: | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | PF15DR | Pin state | R | See table 21.10. |
| 14 | PF14DR | Pin state | R | |
| 13 | PF13DR | Pin state | R | |
| 12 | PF12DR | Pin state | R | |
| 11 | PF11DR | Pin state | R | |
| 10 | PF10DR | Pin state | R | |
| 9 | PF9DR | Pin state | R | |
| 8 | PF8DR | Pin state | R | |
| 7 | PF7DR | Pin state | R | |
| 6 | PF6DR | Pin state | R | |
| 5 | PF5DR | Pin state | R | |
| 4 | PF4DR | Pin state | R | |
| 3 | PF3DR | Pin state | R | |
| 2 | PF2DR | Pin state | R | |
| 1 | PF1DR | Pin state | R | |
| 0 | PF0DR | Pin state | R | |

**Table 21.10 Port F Data Register L (PFDRL) Read/Write Operations**

- PFDRL Bits 15 to 0

| Pin Function | Read | Write |
|---|---|---|
| General input | Pin state | Ignored (no effect on pin state) |
| ANn input (analog input) | 1 | Ignored (no effect on pin state) |

# Section 22 Flash Memory

This LSI has 256-Kbyte or 128-Kbyte on-chip flash memory. The flash memory has the following features.

## 22.1 Features

- Two flash-memory MATs, with one selected by the mode in which the LSI starts up

  The on-chip flash memory has two memory spaces in the same address space (hereafter referred to as memory MATs). The mode setting when the LSI starts up determines the memory MAT that is currently mapped. The MAT can be switched by bank-switching after the LSI has started up.

  — Size of the user MAT, from which booting-up proceeds after a power-on reset in user mode: 256 Kbytes or 128 Kbytes

  — Size of the user boot MAT, from which booting-up proceeds after a power-on reset in user boot mode: 12 Kbytes

- Three on-board programming modes and one off-board programming mode

  **On-board programming modes**

  **Boot Mode**: The on-chip SCI interface is used for programming in this mode. Either the user MAT or user-boot MAT can be programmed, and the bit rate for data transfer between the host and this LSI are automatically adjusted.

  **User Program Mode**: This mode allows programming of the user MAT via any desired interface.

  **User Boot Mode**: This mode allows writing of a user boot program via any desired interface and programming of the user MAT.

  **Off-board programming mode**

  **Programmer Mode:** This mode allows programming of the user MAT and user boot MAT with the aid of a PROM programmer.

- Downloading of an on-chip program to provide an interface for programming/erasure

  This LSI has a dedicated programming/erasing program. After this program has been downloaded to the on-chip RAM, programming or erasing can be performed by setting parameters as arguments. “User branching” is also supported.

— User branching

Programming is performed in 128-byte units. Each round of programming consists of application of the programming pulse, reading for verification, and several other steps. Erasing is performed in block units and each round of erasing consists of several steps. A user-processing routine can be executed between each round of erasing, and making the setting for this is called the addition of a user branch.

- Using on-chip RAM to emulate flash memory

  By laying on-chip RAM over part of the flash memory, flash-memory programming can be emulated in real time.

- Protection modes

  There are two modes of protection: software protection is applied by register settings and hardware protection is applied by the level on the FWE pin. Protection of the flash memory from programming or erasure can be selected.

  When an abnormal state is detected, such as runaway execution of programming/erasing, the protection modes initiate the transition to the error protection state and suspend programming/erasing processing.

- Programming/erasing time

  The time taken to program 128 bytes of flash memory in a single round is $t_P$ ms (typ.), which is equivalent to $t_P$/128 ms per byte. The erasing time is $t_E$s (typ.) per block.

- Number of programming operations

  The flash memory can be programmed up to $N_{WEC}$ times.

- Operating frequency for programming/erasing

  The operating frequency for programming/erasing is a maximum of 40 MHz (Pϕ).

## 22.2 Overview

### 22.2.1 Block Diagram

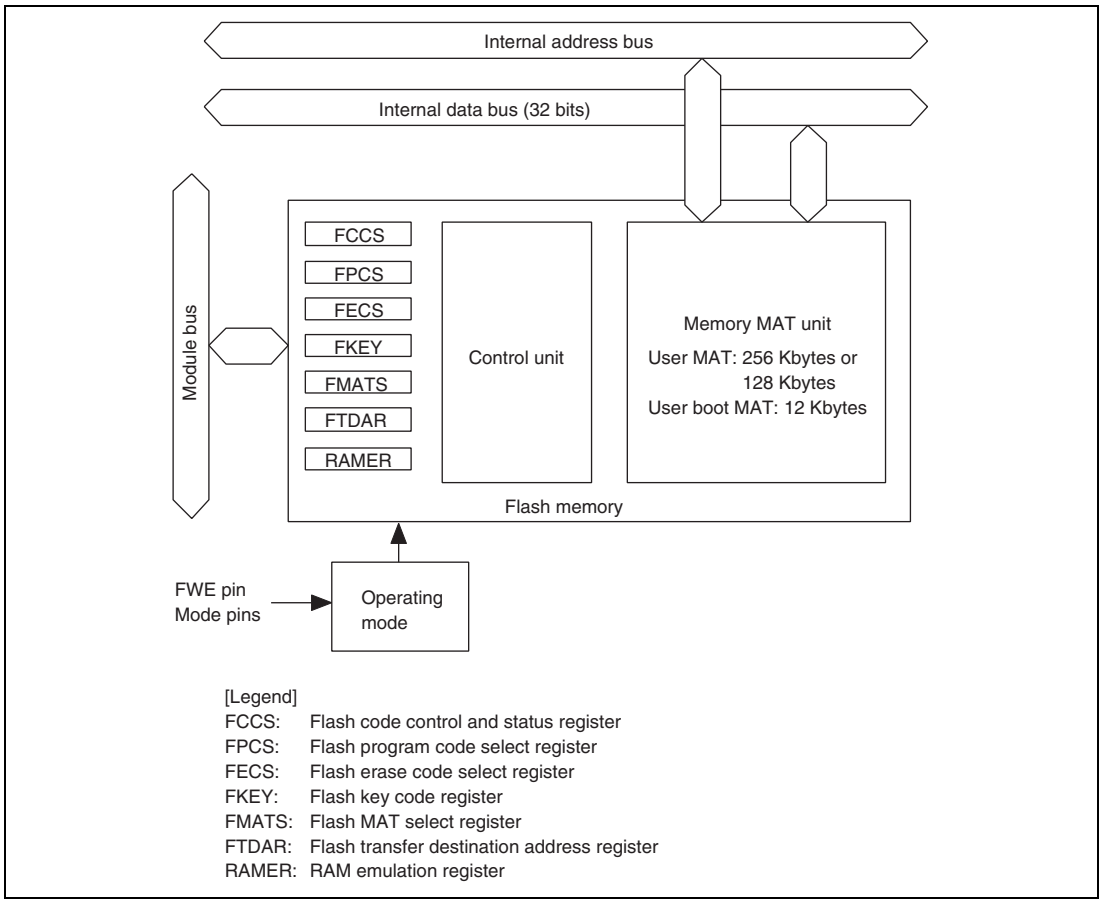



**Figure 22.1 Block Diagram of Flash Memory**

### 22.2.2 Operating Mode

When each mode pin and the FWE pin are set in the reset state and the reset signal is released, the microcomputer enters each operating mode as shown in figure 22.2. For the setting of each mode pin and the FWE pin, see table 22.1.

- Flash memory cannot be read, programmed, or erased in ROM invalid mode. The programming/erasing interface registers cannot be written to. When these registers are read, H'00 is always read.
- Flash memory can be read in user mode, but cannot be programmed or erased.
- Flash memory can be read, programmed, or erased on the board only in user program mode, user boot mode, and boot mode.
- Flash memory can be read, programmed, or erased by means of the PROM programmer in programmer mode.

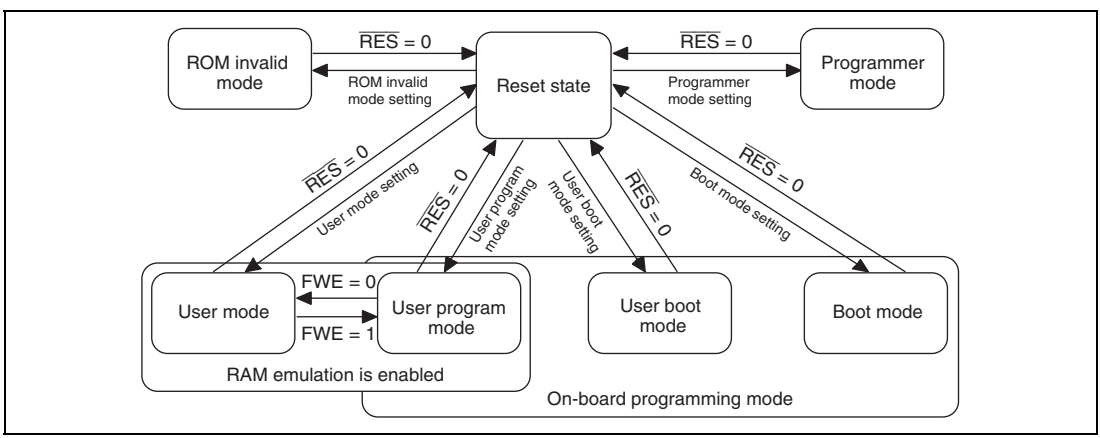



**Figure 22.2 Mode Transition of Flash Memory**

**Table 22.1 (1) Relationship between FWE and MD Pins and Operating Modes (SH7131/SH7136)**

| Pin | Reset State | User Mode | User Program Mode | Boot Mode | Programmer Mode |
|---|---|---|---|---|---|
| $\overline{\text{RES}}$ | 0 | 1 | 1 | 1 | Setting value depends on the condition of the specialized PROM programmer. |
| FWE | 0/1 | 0 | 1 | 1 | |
| MD1 | 0/1 | 1 | 1 | 0 | |

Note: External bus extended mode and user boot mode are not supported by the SH7131 and SH7136.

**Table 22.1 (2) Relationship between FWE and MD Pins and Operating Modes (SH7132/SH7137)**

| Pin | Reset State | ROM Invalid Mode | User Mode | User Program Mode | User Boot Mode | Boot Mode | Programmer Mode |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RES}}$ | 0 | 1 | 1 | 1 | 1 | 1 | Setting value depends on the condition of the specialized PROM programmer. |
| FWE | 0/1 | 0 | 0 | 1 | 1 | 1 | |
| MD0 | 0/1 | 0*[1] | 0/1*[2] | 0/1*[2] | 1 | 0 | |
| MD1 | 0/1 | 0 | 1 | 1 | 0 | 0 | |

Notes: 1. MD0 = 0: 8-bit external bus
2. MD0 = 0: External bus can be used, MD0 = 1: Single-chip mode (external bus cannot be used)

### 22.2.3 Mode Comparison

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and programmer mode is shown in table 22.2.

**Table 22.2 Comparison of Programming Modes**

| | Boot Mode | User Program Mode | User Boot Mode | Programmer Mode |
|---|---|---|---|---|
| Programming/ erasing environment | On-board programming | On-board programming | On-board programming | Off-board programming |
| Programming/ erasing enable MAT | User MAT<br>User boot MAT | User MAT | User MAT | User MAT<br>User boot MAT |
| Programming/ erasing control | Command method | Programming/ erasing interface | Programming/ erasing interface | — |
| All erasure | Possible (Automatic) | Possible | Possible | Possible (Automatic) |
| Block division erasure | Possible*[1] | Possible | Possible | Not possible |
| Program data transfer | From host via SCI | From optional device via RAM | From optional device via RAM | Via programmer |
| User branch function | Not possible | Possible | Possible | Not possible |
| RAM emulation | Not possible | Possible | Not possible | Not possible |
| Reset initiation MAT | Embedded program storage MAT | User MAT | User boot MAT*[2] | Embedded program storage MAT |
| Transition to user mode | Mode setting change and reset | FWE setting change | Mode setting change and reset | — |

Notes: 1. All-erasure is performed. After that, the specified block can be erased.
2. Initiation starts from the embedded program storage MAT. After checking the flash-memory related registers, initiation starts from the reset vector of the user MAT.

- The user boot MAT can be programmed or erased only in boot mode and programmer mode.
- The user MAT and user boot MAT are all erased in boot mode. Then, the user MAT and user boot MAT can be programmed by means of the command method. However, the contents of the MAT cannot be read until this state.
  Only user boot MAT is programmed and the user MAT is programmed in user boot mode or only user MAT is programmed because user boot mode is not used.
- In user boot mode, the boot operation of the optional interface can be performed by a mode pin setting different from user program mode.

### 22.2.4 Flash Memory Configuration

This LSI's flash memory is configured by the 256-Kbyte or 128-Kbyte user MAT and 12-Kbyte user boot MAT.

The start address is allocated to the same address in the user MAT and user boot MAT. Therefore, when the program execution or data access is performed between the two MATs, the MAT must be switched by using FMATS.

The user MAT or user boot MAT can be read in all modes if it is in ROM valid mode. However, the user boot MAT can be programmed only in boot mode and programmer mode.

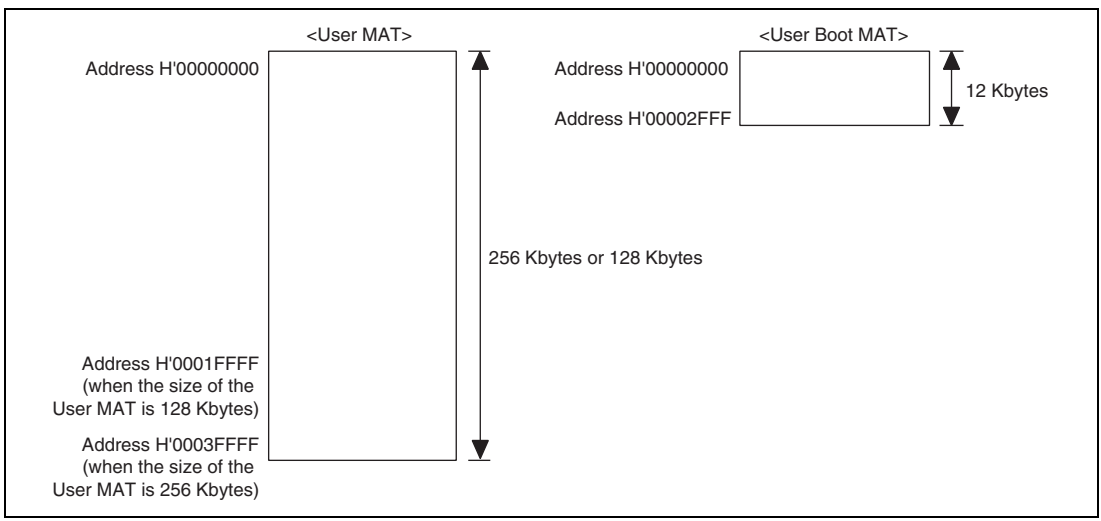



**Figure 22.3 Flash Memory Configuration**

The user MAT and user boot MAT have different memory sizes. Do not access a user boot MAT that is 12 Kbytes or more. When a user boot MAT exceeding 12 Kbytes is read from, an undefined value is read.

### 22.2.5 Block Division

The user MAT is divided into 64 Kbytes (256-Kbyte version: three blocks, 128-Kbyte version: one block), 32 Kbytes (one block), and 4 Kbytes (eight blocks) as shown in figure 22.4. The user MAT can be erased in this divided-block units and the erase-block number of EB0 to EB11 is specified when erasing.

The RAM emulation can be performed in the eight blocks of 4 Kbytes.

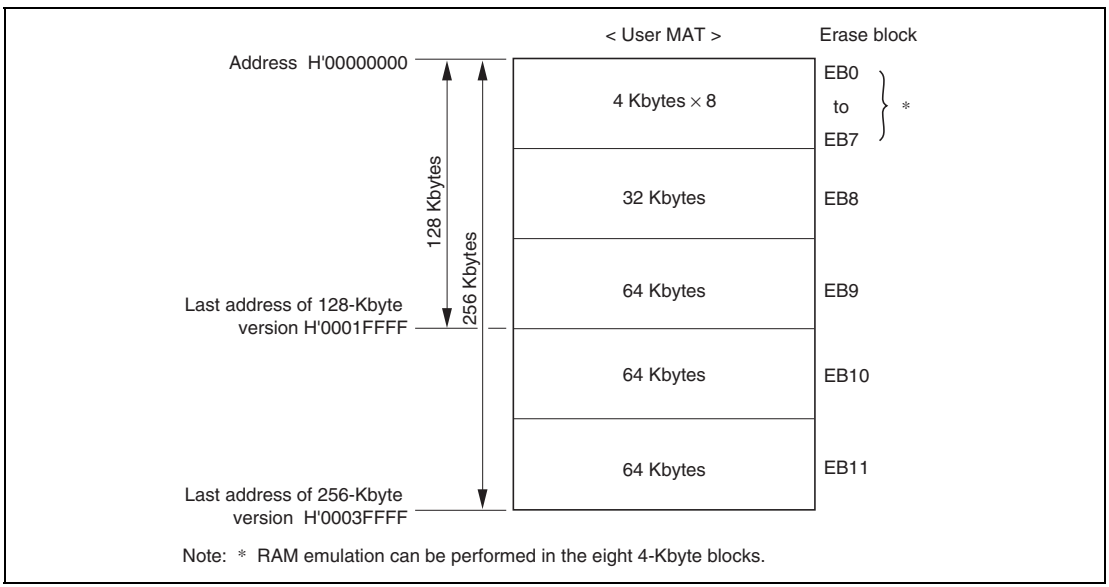



**Figure 22.4 Block Division of User MAT**

### 22.2.6 Programming/Erasing Interface

Programming/erasing is executed by downloading the on-chip program to the on-chip RAM and specifying the program address/data and erase block by using the interface registers/parameters.

The procedure program is made by the user in user program mode and user boot mode. The overview of the procedure is as follows. For details, see section 22.5.2, User Program Mode.

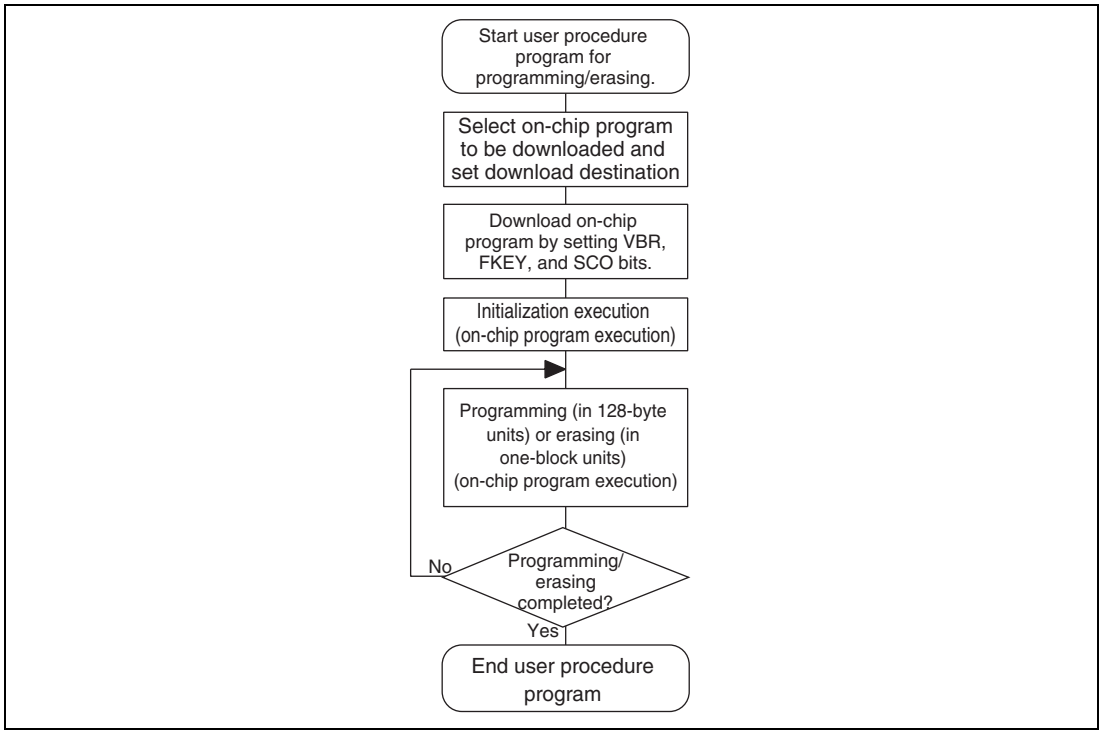



**Figure 22.5 Overview of User Procedure Program**

(1) Selection of On-Chip Program to be Downloaded and Setting of Download Destination

This LSI has programming/erasing programs and they can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by setting the corresponding bits in the programming/erasing interface registers. The download destination can be specified by FTDAR.

(2) Download of On-Chip Program

The on-chip program is automatically downloaded by clearing VBR of the CPU to H'84000000 and then setting the SCO bit in the flash code control and status register (FCCS) and the flash key code register (FKEY), which are programming/erasing interface registers.

The user MAT is replaced to the embedded program storage area when downloading. Since the flash memory cannot be read when programming/erasing, the procedure program, which is working from download to completion of programming/erasing, must be executed in a space other than the flash memory to be programmed/erased (for example, on-chip RAM).

Since the result of download is returned to the programming/erasing interface parameters, whether the normal download is executed or not can be confirmed.

Note that VBR can be changed after download is completed.

(3) Initialization of Programming/Erasing

The operating frequency and user branch are set before execution of programming/erasing. The user branch destination must be in an area other than the user MAT area which is in the middle of programming and the area where the on-chip program is downloaded. These settings are performed by using the programming/erasing interface parameters.

(4) Programming/Erasing Execution

To program or erase, the FWE pin must be brought high and user program mode must be entered.

The program data/programming destination address is specified in 128-byte units when programming.

The block to be erased is specified in erase-block units when erasing.

These specifications are set by using the programming/erasing interface parameters and the on-chip program is initiated. The on-chip program is executed by using the JSR or BSR instruction to perform the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameters.

The area to be programmed must be erased in advance when programming flash memory.

There are limitations and notes on the interrupt processing during programming/erasing. For details, see section 22.8.2, Interrupts during Programming/Erasing.

(5) When Programming/Erasing is Executed Consecutively

When the processing is not ended by the 128-byte programming or one-block erasure, the program address/data and erase-block number must be updated and consecutive programming/erasing is required.

Since the downloaded on-chip program is left in the on-chip RAM after the processing, download and initialization are not required when the same processing is executed consecutively.

## 22.3 Input/Output Pins

Flash memory is controlled by the pins as shown in table 22.3.

**Table 22.3 Pin Configuration**

| Name | Pin Name | Input/Output | Function |
|---|---|---|---|
| Power-on reset | $\overline{\text{RES}}$ | Input | Reset |
| Flash programming enable | FWE | Input | Hardware protection when programming flash memory |
| Mode 1 | MD1 | Input | Sets operating mode of this LSI |
| Mode 0* | MD0 | Input | Sets operating mode of this LSI |
| Transmit data | TXD1 (PA4) | Output | Serial transmit data output (used in boot mode) |
| Receive data | RXD1 (PA3) | Input | Serial receive data input (used in boot mode) |

Note: * The SH7131 and SH7136 do not have the MD0 pin.

## 22.4 Register Descriptions

### 22.4.1 Registers

The registers/parameters which control flash memory when the on-chip flash memory is valid are shown in table 22.4.

There are several operating modes for accessing flash memory, for example, read mode/program mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/parameters are allocated for each operating mode and MAT selection. The correspondence of operating modes and registers/parameters for use is shown in table 22.5.

**Table 22.4 (1) Register Configuration**

| Register Name | Abbreviation*[4] | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Flash code control and status register | FCCS | R, W*[1] | H'00*[2]<br>H'80*[2] | H'FFFFCC00 | 8 |
| Flash program code select register | FPCS | R/W | H'00 | H'FFFFCC01 | 8 |
| Flash erase code select register | FECS | R/W | H'00 | H'FFFFCC02 | 8 |
| Flash key code register | FKEY | R/W | H'00 | H'FFFFCC04 | 8 |
| Flash MAT select register | FMATS | R/W | H'00*[3]<br>H'AA*[3] | H'FFFFCC05 | 8 |
| Flash transfer destination address register | FTDAR | R/W | H'00 | H'FFFFCC06 | 8 |
| RAM emulation register | RAMER | R/W | H'0000 | H'FFFFF108 | 16 |

Notes: 1. The bits except the SCO bit are read-only bits. The SCO bit is a programming-only bit. (The value that can be read is always 0.)
2. The initial value of the FWE bit is 0 when the FWE pin goes low.
The initial value of the FWE bit is 1 when the FWE pin goes high.
3. The initial value at initiation in user mode or user program mode is H'00.
The initial value at initiation in user boot mode is H'AA.
4. All registers except for RAMER can be accessed only in bytes.
RAMER can be accessed in bytes or words.

**Table 22.4 (2) Parameter Configuration**

| Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Download pass/fail result | DPFR | R/W | Undefined | On-chip RAM* | 8, 16, 32 |
| Flash pass/fail result | FPFR | R/W | Undefined | R0 of CPU | 8, 16, 32 |
| Flash multipurpose address area | FMPAR | R/W | Undefined | R5 of CPU | 8, 16, 32 |
| Flash multipurpose data destination area | FMPDR | R/W | Undefined | R4 of CPU | 8, 16, 32 |
| Flash erase block select | FEBS | R/W | Undefined | R4 of CPU | 8, 16, 32 |
| Flash program and erase frequency control | FPEFEQ | R/W | Undefined | R4 of CPU | 8, 16, 32 |
| Flash user branch address set parameter | FUBRA | R/W | Undefined | R5 of CPU | 8, 16, 32 |

Note: * One byte of the start address in the on-chip RAM area specified by FTDAR is valid.

**Table 22.5 Register/Parameter and Target Mode**

| | | Download | Initialization | Programming | Erasure | Read | RAM Emulation |
|---|---|---|---|---|---|---|---|
| Programming/ erasing interface registers | FCCS | √ | — | — | — | — | — |
| | FPCS | √ | — | — | — | — | — |
| | PECS | √ | — | — | — | — | — |
| | FKEY | √ | — | √ | √ | — | — |
| | FMATS | — | — | √*1 | √*1 | √*2 | — |
| | FTDAR | √ | — | — | — | — | — |
| Programming/ erasing interface parameters | DPFR | √ | — | — | — | — | — |
| | FPFR | — | √ | √ | √ | — | — |
| | FPEFEQ | — | √ | — | — | — | — |
| | FUBRA | — | √ | — | — | — | — |
| | FMPAR | — | — | √ | — | — | — |
| | FMPDR | — | — | √ | — | — | — |
| | FEBS | — | — | — | √ | — | — |
| RAM emulation | RAMER | — | — | — | — | — | √ |

Notes: 1. The setting is required when programming or erasing user MAT in user boot mode.
2. The setting may be required according to the combination of initiation mode and read target MAT.

### 22.4.2 Programming/Erasing Interface Registers

The programming/erasing interface registers are as described below. They are all 8-bit registers that can be accessed in bytes.

(1) Flash Code Control and Status Register (FCCS)

FCCS is configured by bits which request the monitor of the FWE pin state and error occurrence during programming or erasing flash memory and the download of the on-chip program.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FWE | MAT | - | FLER | - | - | - | SCO |
| Initial value: | 1/0 | 1/0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | (R)/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | FWE | 1/0 | R | Flash Programming Enable<br>Monitors the level, which is input to the FWE pin that performs hardware protection of the flash memory programming or erasing. The initial value is 0 or 1 according to the FWE pin state.<br>0: When the FWE pin goes low (in hardware protection state)<br>1: When the FWE pin goes high |
| 6 | MAT | 1/0 | R | MAT Bit<br>Indicates whether the user MAT or user boot MAT is selected.<br>0: User MAT is selected<br>1: User boot MAT is selected |
| 5 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | FLER | 0 | R | Flash Memory Error<br>Indicates an error occurs during programming and erasing flash memory.<br>When FLER is set to 1, flash memory enters the error protection state.<br>When FLER is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to flash memory, the reset signal must be released after the reset period of 100 μs, which is longer than normal.<br>0: Flash memory operates normally<br>Programming/erasing protection for flash memory (error protection) is invalid.<br>[Clearing condition]<br>At a power-on reset<br>1: Indicates an error occurs during programming/erasing flash memory.<br>Programming/erasing protection for flash memory (error protection) is valid.<br>[Setting condition]<br>See section 22.6.3, Error Protection. |
| 3 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | SCO | 0 | (R)/W | Source Program Copy Operation<br>Requests the on-chip programming/erasing program to be downloaded to the on-chip RAM.<br>When this bit is set to 1, the on-chip program which is selected by FPCS/FECS is automatically downloaded in the on-chip RAM area specified by FTDAR.<br>In order to set this bit to 1, RAM emulation state must be canceled, H'A5 must be written to FKEY, and this operation must be in the on-chip RAM.<br>Four NOP instructions must be executed immediately after setting this bit to 1.<br>For interrupts during download, see section 22.8.2, Interrupts during Programming/Erasing. For the download time, see section 22.8.3, Other Notes.<br>Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1.<br>Download by setting the SCO bit to 1 requires a special interrupt processing that performs bank switching to the on-chip program storage area. Therefore, before issuing a download request (SCO = 1), set VBR to H'84000000. Otherwise, the CPU gets out of control. Once download end is confirmed, VBR can be changed to any other value.<br>The mode in which the FWE pin is high must be used when using the SCO function.<br>0: Download of the on-chip programming/erasing program to the on-chip RAM is not executed.<br>[Clearing condition]<br>When download is completed<br>1: Request that the on-chip programming/erasing program is downloaded to the on-chip RAM is generated<br>[Setting conditions]<br>When all of the following conditions are satisfied and 1 is written to this bit<br>• FKEY is written to H'A5<br>• During execution in the on-chip RAM<br>• Not in RAM emulation mode (RAMS in RAMCR = 0) |

(2) Flash Program Code Select Register (FPCS)

FPCS selects the on-chip programming program to be downloaded.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | PPVS |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | PPVS | 0 | R/W | Program Pulse Single<br>Selects the programming program.<br>0: On-chip programming program is not selected<br>[Clearing condition]<br>When transfer is completed<br>1: On-chip programming program is selected |

(3) Flash Erase Code Select Register (FECS)

FECS selects download of the on-chip erasing program.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | EPVB |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | EPVB | 0 | R/W | Erase Pulse Verify Block<br>Selects the erasing program.<br>0: On-chip erasing program is not selected<br>[Clearing condition]<br>When transfer is completed<br>1: On-chip erasing program is selected |

(4) Flash Key Code Register (FKEY)

FKEY is a register for software protection that enables download of the on-chip program and programming/erasing of flash memory. Before setting the SCO bit to 1 in order to download the on-chip program or executing the downloaded programming/erasing program, these processing cannot be executed if the key code is not written.

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | K[7:0] | All 0 | R/W | Key Code<br>Only when H'A5 is written, writing to the SCO bit is valid. When a value other than H'A5 is written to FKEY, 1 cannot be written to the SCO bit. Therefore downloading to the on-chip RAM cannot be executed.<br>Only when H'5A is written, programming/erasing of flash memory can be executed. Even if the on-chip programming/erasing program is executed, flash memory cannot be programmed or erased when a value other than H'5A is written to FKEY.<br>H'A5: Writing to the SCO bit is enabled (The SCO bit cannot be set by a value other than H'A5.)<br>H'5A: Programming/erasing is enabled (A value other than H'5A enables software protection state.)<br>H'00: Initial value |

(5) Flash MAT Select Register (FMATS)

FMATS specifies whether user MAT or user boot MAT is selected.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MS7 | MS6 | MS5 | MS4 | MS3 | MS2 | MS1 | MS0 |
| Initial value: | 0/1 | 0 | 0/1 | 0 | 0/1 | 0 | 0/1 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | MS7 | 0/1 | R/W | MAT Select |
| 6 | MS6 | 0 | R/W | These bits are in user-MAT selection state when a value other than H'AA is written and in user-boot-MAT selection state when H'AA is written. |
| 5 | MS5 | 0/1 | R/W | |
| 4 | MS4 | 0 | R/W | |
| 3 | MS3 | 0/1 | R/W | The MAT is switched by writing a value in FMATS with the on-chip RAM instrunction. |
| 2 | MS2 | 0 | R/W | When the MAT is switched, follow section 22.8.1, Switching between User MAT and User Boot MAT. (The user boot MAT cannot be programmed in user program mode if user boot MAT is selected by FMATS. The user boot MAT must be programmed in boot mode or in programmer mode.) |
| 1 | MS1 | 0/1 | R/W | |
| 0 | MS0 | 0 | R/W | |
| | | | | H'AA: The user boot MAT is selected (in user-MAT selection state when the value of these bits are other than H'AA)<br>Initial value when these bits are initiated in user boot mode. |
| | | | | H'00: Initial value when these bits are initiated in a mode except for user boot mode (in user-MAT selection state) |
| | | | | [Programmable condition]<br>These bits are in the execution state in the on-chip RAM. |

(6) Flash Transfer Destination Address Register (FTDAR)

FTDAR specifies the on-chip RAM address to which the on-chip program is downloaded.

Make settings for FTDAR before writing 1 to the SCO bit in FCCS. The initial value is H'00 which points to the start address (H'FFFF9000) in on-chip RAM.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TDER | TDA[6:0] | | | | | | |
| 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 | TDER | 0 | R/W | Transfer Destination Address Setting Error<br>This bit is set to 1 when there is an error in the download start address set by bits 6 to 0 (TDA6 to TDA0). Whether the address setting is erroneous or not is tested by checking whether the setting of TDA6 to TDA0 is in the range of H'00 to H'04 after setting the SCO bit in FCCS to 1 and performing download. Before setting the SCO bit to 1 be sure to set the FTDAR value between H'00 to H'04 as well as clearing this bit to 0.<br>0: Setting of TDA6 to TDA0 is normal<br>1: Setting of TDER and TDA6 to TDA0 is H'05 to H'FF and download has been aborted |
| 6 to 0 | TDA[6:0] | All 0 | R/W | Transfer Destination Address<br>These bits specify the download start address. A value from H'00 to H'04 can be set to specify the download start address in on-chip RAM in 2-Kbyte units.<br>A value from H'05 to H'7F cannot be set. If such a value is set, the TDER bit (bit 7) in this register is set to 1 to prevent download from being executed.<br>H'00: Download start address is set to H'FFFF9000<br>H'01: Download start address is set to H'FFFF9800<br>H'02: Download start address is set to H'FFFFA000<br>H'03: Download start address is set to H'FFFFA800<br>H'04: Download start address is set to H'FFFFB000<br>H'05 to H'7F: Setting prohibited. If this value is set, the TDER bit (bit 7) is set to 1 to abort the download processing. |

### 22.4.3 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, user branch destination address, storage place for program data, programming destination address, and erase block and exchanges the processing result for the downloaded on-chip program. This parameter uses the general registers of the CPU (R4, R5, and R0) or the on-chip RAM area. The initial value is undefined.

At download all CPU registers are stored, and at initialization or when the on-chip program is executed, CPU registers except for R0 are stored. The return value of the processing result is written in R0. Since the stack area is used for storing the registers or as a work area, the stack area must be saved at the processing start. (The maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameters are used in the following four items.

1. Download control
2. Initialization before programming or erasing
3. Programming
4. Erasing

These items use different parameters. The correspondence table is shown in table 22.6.

The processing results of initialization, programming, and erasing are returned, but the bit contents have different meanings according to the processing program. See the description of FPFR for each processing.

**Table 22.6 Usable Parameters and Target Modes**

| Name of Parameter | Abbreviation | Download | Initialization | Programming | Erasure | R/W | Initial Value | Allocation |
|---|---|---|---|---|---|---|---|---|
| Download pass/fail result | DPFR | √ | — | — | — | R/W | Undefined | On-chip RAM* |
| Flash pass/fail result | FPFR | — | √ | √ | √ | R/W | Undefined | R0 of CPU |
| Flash programming/ erasing frequency control | FPEFEQ | — | √ | — | — | R/W | Undefined | R4 of CPU |
| Flash user branch address set | FUBRA | — | √ | — | — | R/W | Undefined | R5 of CPU |
| Flash multipurpose address area | FMPAR | — | — | √ | — | R/W | Undefined | R5 of CPU |
| Flash multipurpose data destination area | FMPDR | — | — | √ | — | R/W | Undefined | R4 of CPU |
| Flash erase block select | FEBS | — | — | — | √ | R/W | Undefined | R4 of CPU |

Note: * One byte of start address of download destination specified by FTDAR

(1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-chip RAM area to be downloaded is the area as much as 3 Kbytes starting from the start address specified by FTDAR. For the address map of the on-chip RAM, see figure 22.10.

The download control is set by using the programming/erasing interface registers. The return value is given by the DPFR parameter.

(a) Download pass/fail result parameter (DPFR: one byte of start address of on-chip RAM specified by FTDAR)

This parameter indicates the return value of the download result. The value of this parameter can be used to determine if downloading is executed or not. Since the confirmation whether the SCO bit is set to 1 is difficult, the certain determination must be performed by setting one byte of the start address of the on-chip RAM area specified by FTDAR to a value other than the return value of download (for example, H'FF) before the download start (before setting the SCO bit to 1). For the checking method of download results, see section 22.5.2 (2), Programming Procedure in User Program Mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | SS | FK | SF |
| Initial value: | - | - | - | - | - | - | - | - |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 3 | — | Undefined | R/W | Unused<br>Return 0. |
| 2 | SS | Undefined | R/W | Source Select Error Detect<br>The on-chip program which can be downloaded can be specified as only one type. When more than two types of the program are selected, the program is not selected, or the program is selected without mapping, an error occurs.<br>0: Download program can be selected normally<br>1: Download error occurs (Multi-selection or program which is not mapped is selected) |
| 1 | FK | Undefined | R/W | Flash Key Register Error Detect<br>Returns the check result whether the value of FKEY is set to H'A5.<br>0: FKEY setting is normal (FKEY = H'A5)<br>1: FKEY setting is abnormal (FKEY = value other than H'A5) |
| 0 | SF | Undefined | R/W | Success/Fail<br>Returns the result whether download has ended normally or not.<br>0: Downloading on-chip program has ended normally (no error)<br>1: Downloading on-chip program has ended abnormally (error occurs) |

(2) Programming/Erasing Initialization

The on-chip programming/erasing program to be downloaded includes the initialization program.

The specified period pulse must be applied when programming or erasing. The specified pulse width is made by the method in which wait loop is configured by the CPU instruction. The operating frequency of the CPU must be set. Since the user branch function is supported, the user branch destination address must be set.

The initial program is set as a parameter of the programming/erasing program which has downloaded these settings.

(2.1) Flash programming/erasing frequency parameter (FPEFEQ: general register R4 of CPU)

This parameter sets the operating frequency of the CPU.

For the range of the operating frequency of this LSI, see section 26.3.1, Clock Timing.

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

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F15 | F14 | F13 | F12 | F11 | F10 | F9 | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | — | Undefined | R/W | Unused<br>Return 0. |
| 15 to 0 | F15 to F0 | Undefined | R/W | Frequency Set<br>Set the operating frequency of the CPU. The setting value must be calculated as the following methods.<br>1. The operating frequency which is shown in MHz units must be rounded in a number to three decimal places and be shown in a number of two decimal places.<br>2. The centuplicated value is converted to the binary digit and is written to the FPEFEQ parameter (general register R4). For example, when the operating frequency of the CPU is 28.882 MHz, the value is as follows.<br>— The number to three decimal places of 28.882 is rounded and the value is thus 28.88.<br>— The formula that 28.88 × 100 = 2888 is converted to the binary digit and B'0000, B'1011, B'0100, B'1000 (H'0B48) is set to B'R4. |

(2.2) Flash user branch address setting parameter (FUBRA: general register R5 of CPU)

This parameter sets the user branch destination address. The user program which has been set can be executed in specified processing units when programming and erasing.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | UA31 | UA30 | UA29 | UA28 | UA27 | UA26 | UA25 | UA24 | UA23 | UA22 | UA21 | UA20 | UA19 | UA18 | UA17 | UA16 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | UA15 | UA14 | UA13 | UA12 | UA11 | UA10 | UA9 | UA8 | UA7 | UA6 | UA5 | UA4 | UA3 | UA2 | UA1 | UA0 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | UA31 to UA0 | Undefined | R/W | User Branch Destination Address<br>When the user branch is not required, address 0 (H'00000000) must be set.<br>The user branch destination must be an area other than the flash memory, an area other than the RAM area in which on-chip program has been transferred, or the external bus space.<br>Note that the CPU must not branch to an area without the execution code and get out of control. The on-chip program download area and stack area must not be overwritten. If CPU runaway occurs or the download area or stack area is overwritten, the value of flash memory cannot be guaranteed.<br>The download of the on-chip program, initialization, initiation of the programming/erasing program must not be executed in the processing of the user branch destination. Programming or erasing cannot be guaranteed when returning from the user branch destination. The program data which has already been prepared must not be programmed.<br>Store general registers R8 to R15. General registers R0 to R7 are available without storing them.<br>Moreover, the programming/erasing interface registers must not be written to or RAM emulation mode must not be entered in the processing of the user branch destination.<br>After the processing of the user branch has ended, the programming/erasing program must be returned to by using the RTS instruction.<br>For the execution intervals of the user branch processing, see note 2 (User branch processing intervals) in section 22.8.3, Other Notes. |

(2.3) Flash pass/fail result parameter (FPFR: general register R0 of CPU)

This parameter indicates the return value of the initialization result.

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 3 | — | Undefined | R/W | Unused<br>Return 0. |
| 2 | BR | Undefined | R/W | User Branch Error Detect<br>Returns the check result whether the specified user branch destination address is in the area other than the storage area of the programming/erasing program which has been downloaded.<br>0: User branch address setting is normal<br>1: User branch address setting is abnormal |
| 1 | FQ | Undefined | R/W | Frequency Error Detect<br>Returns the check result whether the specified operating frequency of the CPU is in the range of the supported operating frequency.<br>0: Setting of operating frequency is normal<br>1: Setting of operating frequency is abnormal |
| 0 | SF | Undefined | R/W | Success/Fail<br>Indicates whether initialization is completed normally.<br>0: Initialization has ended normally (no error)<br>1: Initialization has ended abnormally (error occurs) |

(3) Programming Execution

When flash memory is programmed, the programming destination address and programming data on the user MAT must be passed to the programming program in which the program data is downloaded.

1. The start address of the programming destination on the user MAT is set in general register R5 of the CPU. This parameter is called FMPAR (flash multipurpose address area parameter).
   Since the program data is always in 128-byte units, the lower eight bits (MOA7 to MOA0) must be H'00 or H'80 as the boundary of the programming start address on the user MAT.
2. The program data for the user MAT must be prepared in the consecutive area. The program data must be in the consecutive space, which can be accessed by using the MOV.B instruction of the CPU, and is not the flash memory space.
   When data to be programmed does not satisfy 128 bytes, the 128-byte program data must be prepared by embedding the dummy code (H'FF).
   The start address of the area in which the prepared program data is stored must be set in general register R4. This parameter is called FMPDR (flash multipurpose data destination area parameter).
   For details on the programming procedure, see section 22.5.2, User Program Mode.

(3.1) Flash multipurpose address area parameter (FMPAR: general register R5 of CPU)

This parameter indicates the start address of the programming destination on the user MAT.

When an address in an area other than the flash memory space is set, an error occurs.

The start address of the programming destination must be at the 128-byte boundary. If this boundary condition is not satisfied, an error occurs. The error occurrence is indicated by the WA bit (bit 1) in FPFR.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MOA31 | MOA30 | MOA29 | MOA28 | MOA27 | MOA26 | MOA25 | MOA24 | MOA23 | MOA22 | MOA21 | MOA20 | MOA19 | MOA18 | MOA17 | MOA16 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MOA15 | MOA14 | MOA13 | MOA12 | MOA11 | MOA10 | MOA9 | MOA8 | MOA7 | MOA6 | MOA5 | MOA4 | MOA3 | MOA2 | MOA1 | MOA0 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | MOA31 to MOA0 | Undefined | R/W | MOA31 to MOA0<br>Store the start address of the programming destination on the user MAT. The consecutive 128-byte programming is executed starting from the specified start address of the user MAT. The MOA6 to MOA0 bits are always 0 because the start address of the programming destination is at the 128-byte boundary. |

(3.2) Flash multipurpose data destination area parameter (FMPDR: general register R4 of CPU)

This parameter indicates the start address in the area, which stores the data to be programmed in the user MAT. When the storage destination of the program data is in flash memory, an error occurs. The error occurrence is indicated by the WD bit (bit 2) in FPFR.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MOD31 | MOD30 | MOD29 | MOD28 | MOD27 | MOD26 | MOD25 | MOD24 | MOD23 | MOD22 | MOD21 | MOD20 | MOD19 | MOD18 | MOD17 | MOD16 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MOD15 | MOD14 | MOD13 | MOD12 | MOD11 | MOD10 | MOD9 | MOD8 | MOD7 | MOD6 | MOD5 | MOD4 | MOD3 | MOD2 | MOD1 | MOD0 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | MOD31 to MOD0 | Undefined | R/W | MOD31 to MOD0<br>Store the start address of the area which stores the program data for the user MAT. The consecutive 128-byte data is programmed to the user MAT starting from the specified start address. |

(3.3) Flash pass/fail result parameter (FPFR: general register R0 of CPU)

This parameter indicates the return value of the program processing result.

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

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | MD | EE | FK | - | WD | WA | SF |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 7 | — | Undefined | R/W | Unused<br>Return 0. |
| 6 | MD | Undefined | R/W | Programming Mode Related Setting Error Detect<br>Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is not entered.<br>When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed with the FWE bit (bit 7) and the FLER bit (bit 4) in FCCS, respectively. For conditions to enter the error protection state, see section 22.6.3, Error Protection.<br>0: FWE and FLER settings are normal (FWE = 1, FLER = 0)<br>1: FWE = 0 or FLER = 1, and programming cannot be performed |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | EE | Undefined | R/W | Programming Execution Error Detect<br>1 is returned to this bit when the specified data could not be written because the user MAT was not erased or when flash-memory related register settings are partially changed on returning from the user branch processing.<br>If this bit is set to 1, there is a high possibility that the user MAT is partially rewritten. In this case, after removing the error factor, erase the user MAT.<br>If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when programming is performed. In this case, both the user MAT and user boot MAT are not rewritten.<br>Programming of the user boot MAT must be executed in boot mode or programmer mode.<br>0: Programming has ended normally<br>1: Programming has ended abnormally (programming result is not guaranteed) |
| 4 | FK | Undefined | R/W | Flash Key Register Error Detect<br>Returns the check result of the value of FKEY before the start of the programming processing.<br>0: FKEY setting is normal (FKEY = H'5A)<br>1: FKEY setting is error (FKEY = value other than H'5A) |
| 3 | — | Undefined | R/W | Unused<br>Return 0. |
| 2 | WD | Undefined | R/W | Write Data Address Error Detect<br>When an address in the flash memory area is specified as the start address of the storage destination of the program data, an error occurs.<br>0: Setting of write data address is normal<br>1: Setting of write data address is abnormal |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | WA | Undefined | R/W | Write Address Error Detect<br>When the following items are specified as the start address of the programming destination, an error occurs.<br>• The programming destination address is an area other than flash memory<br>• The specified address is not at the 128-byte boundary (A6 to A0 are not 0)<br>0: Setting of programming destination address is normal<br>1: Setting of programming destination address is abnormal |
| 0 | SF | Undefined | R/W | Success/Fail<br>Indicates whether the program processing has ended normally or not.<br>0: Programming has ended normally (no error)<br>1: Programming has ended abnormally (error occurs) |

(4) Erasure Execution

When flash memory is erased, the erase-block number on the user MAT must be passed to the erasing program, which is downloaded. This is set to the FEBS parameter (general register R4).

One block is specified from the block number 0 to 15.

For details on the erasing procedure, see section 22.5.2, User Program Mode.

(4.1) Flash erase block select parameter (FEBS: general register R4 of CPU)

This parameter specifies the erase-block number. Several block numbers cannot be specified.

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | Undefined | R/W | Unused<br>Return 0. |
| 7 to 0 | EBS[7:0] | Undefined | R/W | • 256-Kbyte flash memory<br>Set the erase-block number in the range from 0 to 11. 0 corresponds to the EB0 block and 11 corresponds to the EB11 block. An error occurs when a number other than 0 to 11 (H'00 to H'0B) is set.<br>• 128-Kbyte flash memory<br>Set the erase-block number in the range from 0 to 9. 0 corresponds to the EB0 block and 9 corresponds to the EB9 block. An error occurs when a number other than 0 to 9 (H'00 to H'09) is set. |

(4.2) Flash pass/fail result parameter (FPFR: general register R0 of CPU)

This parameter returns the value of the erasing processing result.

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 7 | — | Undefined | R/W | Unused<br>Return 0. |
| 6 | MD | Undefined | R/W | Erasure Mode Related Setting Error Detect<br>Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is not entered.<br>When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed with the FWE bit (bit 7) and the FLER bit (bit 4) in FCCS, respectively. For conditions to enter the error protection state, see section 22.6.3, Error Protection.<br>0: FWE and FLER settings are normal (FWE = 1, FLER = 0)<br>1: FWE = 0 or FLER = 1, and erasure cannot be performed |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | EE | Undefined | R/W | Erasure Execution Error Detect<br>1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed on returning from the user branch processing.<br>If this bit is set to 1, there is a high possibility that the user MAT is partially erased. In this case, after removing the error factor, erase the user MAT.<br>If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and user boot MAT are not erased.<br>Erasure of the user boot MAT must be executed in boot mode or programmer mode.<br>0: Erasure has ended normally<br>1: Erasure has ended abnormally (erasure result is not guaranteed) |
| 4 | FK | Undefined | R/W | Flash Key Register Error Detect<br>Returns the check result of FKEY value before start of the erasing processing.<br>0: FKEY setting is normal (FKEY = H'5A)<br>1: FKEY setting is error (FKEY = value other than H'5A) |
| 3 | EB | Undefined | R/W | Erase Block Select Error Detect<br>Returns the check result whether the specified erase-block number is in the block range of the user MAT.<br>0: Setting of erase-block number is normal<br>1: Setting of erase-block number is abnormal |
| 2, 1 | — | Undefined | R/W | Unused<br>Return 0. |
| 0 | SF | Undefined | R/W | Success/Fail<br>Indicates whether the erasing processing has ended normally or not.<br>0: Erasure has ended normally (no error)<br>1: Erasure has ended abnormally (error occurs) |

### 22.4.4 RAM Emulation Register (RAMER)

When the realtime programming of the user MAT is emulated, RAMER sets the area of the user MAT which is overlapped with a part of the on-chip RAM. The RAM emulation must be executed in user mode or in user program mode.

For the division method of the user-MAT area, see table 22.7. In order to operate the emulation function certainly, the target MAT of the RAM emulation must not be accessed immediately after RAMER is programmed. If it is accessed, the normal access is not guaranteed.

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 3 | RAMS | 0 | R/W | RAM Select<br>Sets whether the user MAT is emulated or not. When RAMS = 1, all blocks of the user MAT are in the programming/erasing protection state.<br>0: Emulation is not selected<br>Programming/erasing protection of all user-MAT blocks is invalid<br>1: Emulation is selected<br>Programming/erasing protection of all user-MAT blocks is valid |
| 2 to 0 | RAM[2:0] | 000 | R/W | User MAT Area Select<br>These bits are used with bit 3 to select the user-MAT area to be overlapped with the on-chip RAM. (See table 22.7.) |

**Table 22.7 Overlapping of RAM Area and User MAT Area**

| RAM Area | Block Name | RAMS | RAM2 | RAM1 | RAM0 |
|---|---|---|---|---|---|
| H'FFFFA000 to H'FFFFAFFF | RAM area (4 Kbytes) | 0 | x | x | x |
| H'00000000 to H'00000FFF | EB0 (4 Kbytes) | 1 | 0 | 0 | 0 |
| H'00001000 to H'00001FFF | EB1 (4 Kbytes) | 1 | 0 | 0 | 1 |
| H'00002000 to H'00002FFF | EB2 (4 Kbytes) | 1 | 0 | 1 | 0 |
| H'00003000 to H'00003FFF | EB3 (4 Kbytes) | 1 | 0 | 1 | 1 |
| H'00004000 to H'00004FFF | EB4 (4 Kbytes) | 1 | 1 | 0 | 0 |
| H'00005000 to H'00005FFF | EB5 (4 Kbytes) | 1 | 1 | 0 | 1 |
| H'00006000 to H'00006FFF | EB6 (4 Kbytes) | 1 | 1 | 1 | 0 |
| H'00007000 to H'00007FFF | EB7 (4 Kbytes) | 1 | 1 | 1 | 1 |

Note: x: Don't care.

## 22.5 On-Board Programming Mode

When the pin is set in on-board programming mode and the reset start is executed, the on-board programming state that can program/erase the on-chip flash memory is entered. On-board programming mode has three operating modes: user program mode, user boot mode, and boot mode.

For details on the pin setting for entering each mode, see table 22.1. For details on the state transition of each mode for flash memory, see figure 22.2.

### 22.5.1 Boot Mode

Boot mode executes programming/erasing user MAT and user boot MAT by means of the control command and program data transmitted from the host using the on-chip SCI. The tool for transmitting the control command and program data must be prepared in the host. The SCI communication mode is set to asynchronous mode. When reset start is executed after this LSI's pin is set in boot mode, the boot program in the microcomputer is initiated. After the SCI bit rate is automatically adjusted, the communication with the host is executed by means of the control command method.

The system configuration diagram in boot mode is shown in figure 22.6. For details on the pin setting in boot mode, see table 22.1. Interrupts are ignored in boot mode so do not generate them.

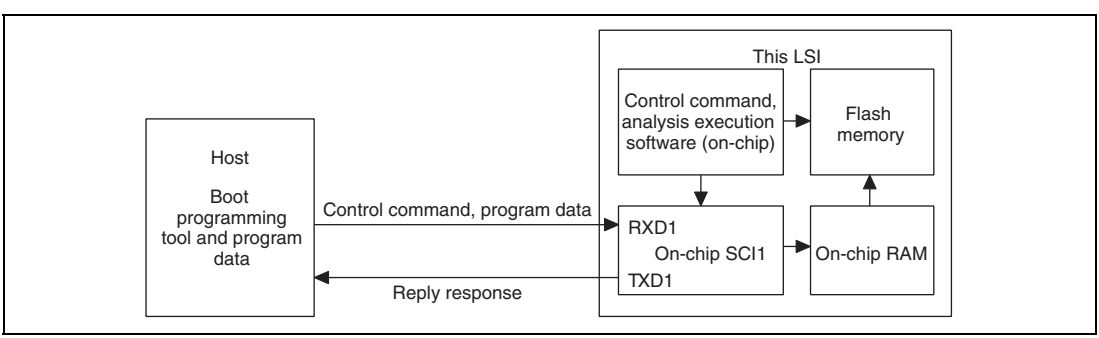



**Figure 22.6 System Configuration in Boot Mode**

(1) SCI Interface Setting by Host

When boot mode is initiated, this LSI measures the low period of asynchronous SCI-communication data (H'00), which is transmitted consecutively by the host. The SCI transmit/receive format is set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate of transmission by the host by means of the measured low period and transmits the bit adjustment end sign (1 byte of H'00) to the host. The host must confirm that this bit adjustment end sign (H'00) has been received normally and transmits 1 byte of H'55 to this LSI. When reception is not executed normally, boot mode is initiated again (reset) and the operation described above must be executed. The bit rate between the host and this LSI is not matched because of the bit rate of transmission by the host and system clock frequency of this LSI. To operate the SCI normally, the transfer bit rate of the host must be set to 9,600 bps or 19,200 bps.

The system clock frequency which can automatically adjust the transfer bit rate of the host and the bit rate of this LSI is shown in table 22.8. Boot mode must be initiated in the range of this system clock. Note that the internal clock division ratio of ×1/3 is not supported in boot mode.

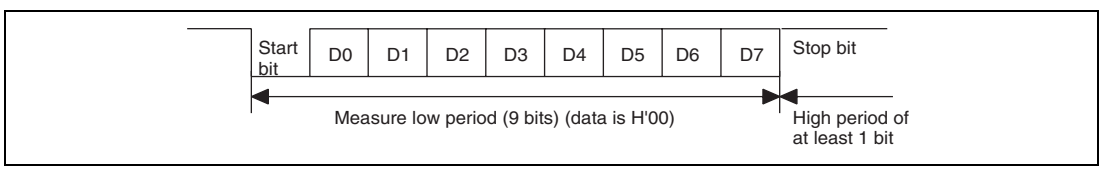



**Figure 22.7 Automatic Adjustment Operation of SCI Bit Rate**

**Table 22.8 Peripheral Clock (Pϕ) Frequency that Can Automatically Adjust Bit Rate of This LSI**

| Host Bit Rate | Peripheral Clock (Pϕ) Frequency Which Can Automatically Adjust LSI's Bit Rate |
|---|---|
| 9,600 bps | 10 to 40 MHz |
| 19,200 bps | 10 to 40 MHz |

Note: The internal clock division ratio of ×1/3 is not supported in boot mode.

(2) State Transition Diagram

Figure 22.8 gives an overview of the state transitions after the chip has been started up in boot mode. For details on boot mode, see section 22.9.1, Specifications of the Standard Serial Communications Interface in Boot Mode.

1. Bit-rate matching

   After the chip has been started up in boot mode, bit-rate matching between the SCI and the host proceeds.

2. Waiting for inquiry and selection commands

   The chip sends the requested information to the host in response to inquiries regarding the size and configuration of the user MAT, start addresses of the MATs, information on supported devices, etc.

3. Automatic erasure of the entire user MAT and user boot MAT

   After all necessary inquiries and selections have been made and the command for transition to the programming/erasure state is sent by the host, the entire user MAT and user boot MAT are automatically erased.

4. Waiting for programming/erasure command

— On receiving the programming selection command, the chip waits for data to be programmed. To program data, the host transmits the programming command code followed by the address where programming should start and the data to be programmed. This is repeated as required while the chip is in the programming-selected state. To terminate programming, H'FFFFFFFF should be transmitted as the first address of the area for programming. This makes the chip return to the programming/erasure command waiting state from the programming data waiting state.

— On receiving the erasure select command, the chip waits for the block number of a block to be erased. To erase a block, the host transmits the erasure command code followed by the number of the block to be erased. This is repeated as required while the chip is in the erasure-selected state. To terminate erasure, H'FF should be transmitted as the block number. This makes the chip return to the programming/erasure command waiting state from the erasure block number waiting state. Erasure should only be executed when a specific block is to be reprogrammed without executing a reset-start of the chip after the flash memory has been programmed in boot mode. If all desired programming is done in a single operation, such erasure processing is not necessary because all blocks are erased before the chip enters the programming/erasure/other command waiting state.

— In addition to the programming and erasure commands, commands for sum checking and blank checking (checking for erasure) of the user MAT and user boot MAT, reading data from the user MAT/user boot MAT, and acquiring current state information are provided.

Note that the command for reading from the user MAT/user boot MAT can only read data that has been programmed after automatic erasure of the entire user MAT and user boot MAT.

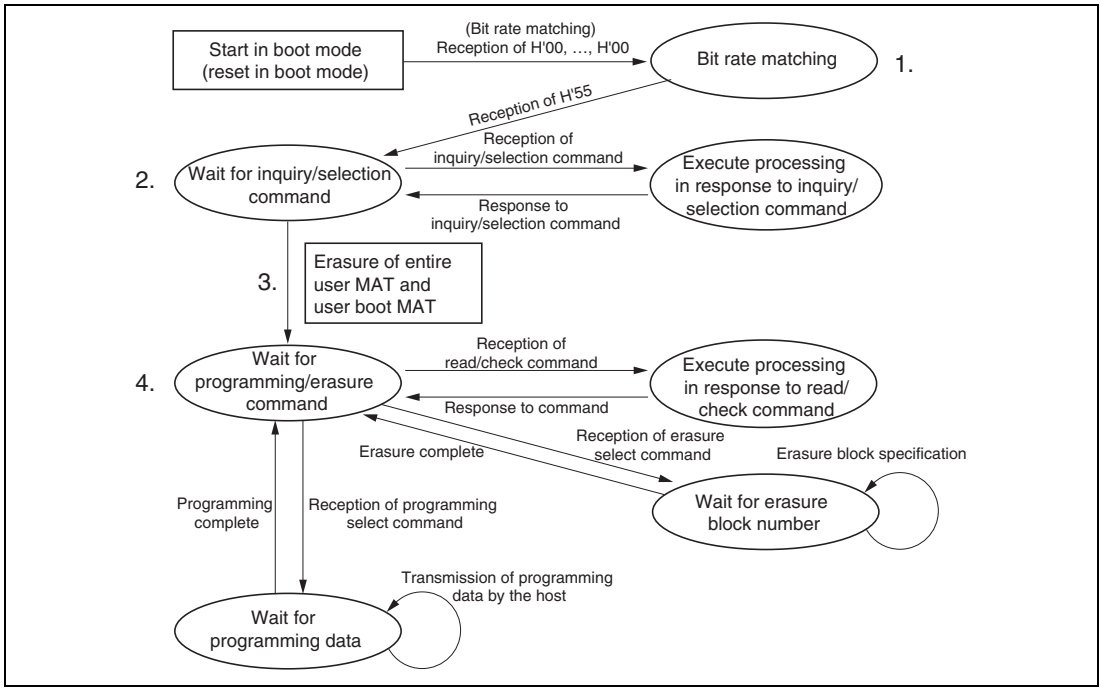



**Figure 22.8 State Transitions in Boot Mode**

### 22.5.2 User Program Mode

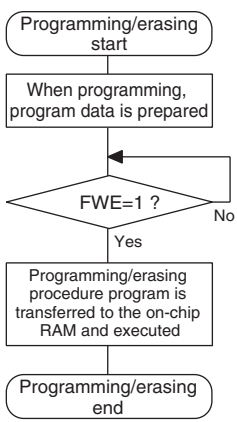
The user MAT can be programmed/erased in user program mode. (The user boot MAT cannot be programmed/erased.)

Programming/erasing is executed by downloading the program in the microcomputer.

The overview flow is shown in figure 22.9.

High voltage is applied to internal flash memory during the programming/erasing processing. Therefore, transition to reset must not be executed. Doing so may cause damage or destroy flash memory. If reset is executed accidentally, the reset signal must be released after the reset input period, which is longer than the normal 100 μs.

For details on the programming procedure, see the description in section 22.5.2 (2), Programming Procedure in User Program Mode. For details on the erasing procedure, see the description in section 22.5.2 (3), Erasing Procedure in User Program Mode.



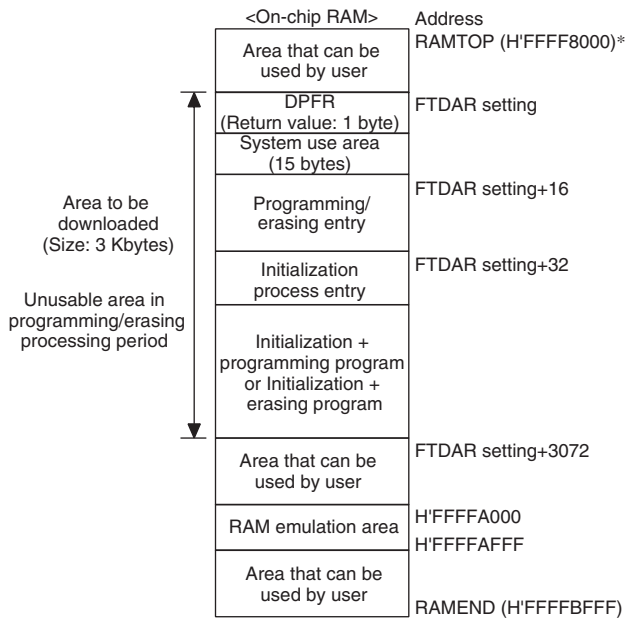

1. RAM emulation mode must be canceled in advance. Download cannot be executed in emulation mode.
2. When the program data is made by means of emulation, the download destination must be changed by FTDAR.
3. Inputting high level to the FWE pin sets the FWE bit to 1.
4. Programming/erasing is executed only in the on-chip RAM. However, if the program data is in a consecutive area and can be accessed by the MOV.B instruction of the CPU like SRAM/ROM, the program data can be in an external space.
5. After programming/erasing is finished, low level must be input to the FWE pin for protection.

**Figure 22.9 Programming/Erasing Overview Flow**

(1) On-Chip RAM Address Map when Programming/Erasing is Executed

Parts of the procedure program that are made by the user, like download request, programming/erasing procedure, and decision of the result, must be executed in the on-chip RAM. All of the on-chip program that is to be downloaded is in on-chip RAM. Note that on-chip RAM must be controlled so that these parts do not overlap.

Figure 22.10 shows the program area to be downloaded.




Note: * H'FFFFA000 on SH7131/SH7132.

**Figure 22.10 RAM Map after Download**

(2) Programming Procedure in User Program Mode

The procedures for download, initialization, and programming are shown in figure 22.11.

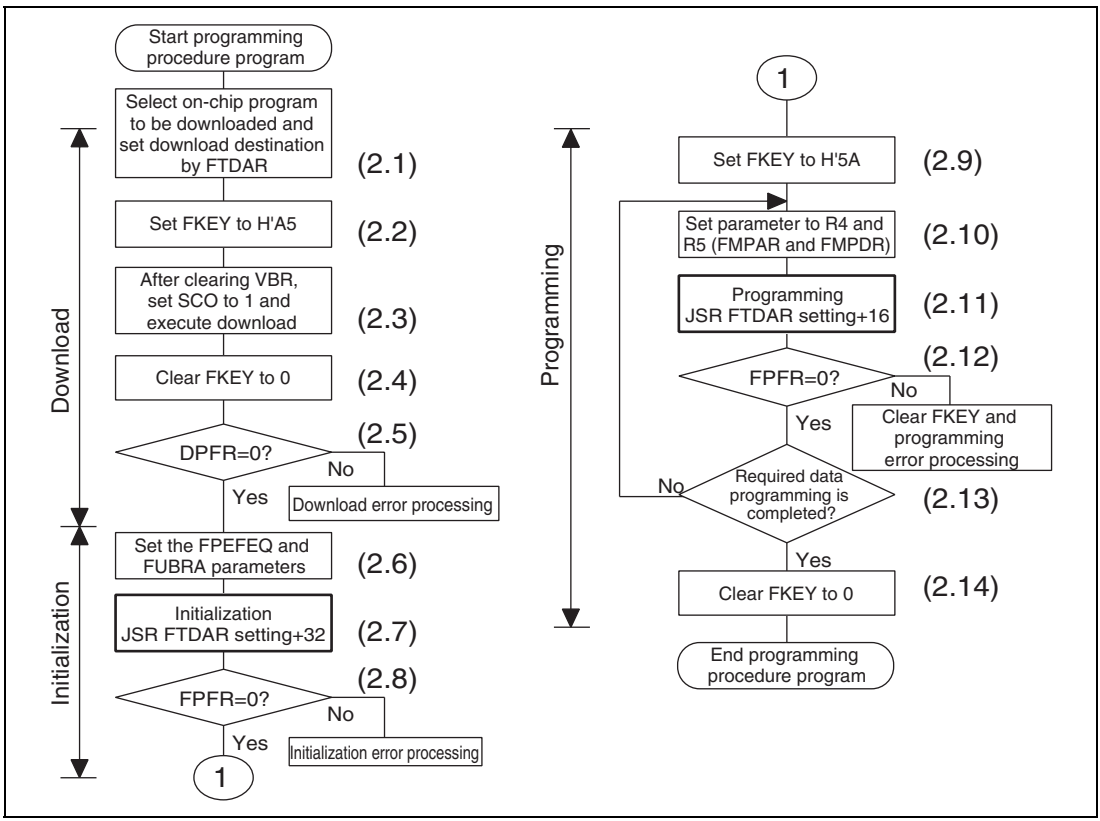



**Figure 22.11 Programming Procedure**

The details of the programming procedure are described below. The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM. Specify 1/4 (initial value) as the frequency division ratios of an internal clock (Iϕ), a bus clock (Bϕ), and a peripheral clock (Pϕ) through the frequency control register (FRQCR).

After the programming/erasing program has been downloaded and the SCO bit is cleared to 0, the setting of the frequency control register (FRQCR) can be changed to the desired value.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 22.9.2, Areas for Storage of the Procedural Program and Data for Programming.

The following description assumes the area to be programmed on the user MAT is erased and program data is prepared in the consecutive area. When erasing has not been executed, carry out erasing before writing.

128-byte programming is performed in one program processing. When more than 128-byte programming is performed, programming destination address/program data parameter is updated in 128-byte units and programming is repeated.

When less than 128-byte programming is performed, data must total 128 bytes by adding the invalid data. If the invalid data to be added is H'FF, the program processing period can be shortened.

(2.1) Select the on-chip program to be downloaded

When the PPVS bit of FPCS is set to 1, the programming program is selected.
Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPFR parameter.
Specify the start address of the download destination by FTDAR.

(2.2) Write H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be written to the SCO bit for a download request.

(2.3) VBR is set to 0 and 1 is written to the SCO bit of FCCS, and then download is executed.

VBR must always be set to H'84000000 before setting the SCO bit to 1.
To write 1 to the SCO bit, the following conditions must be satisfied.

- RAM emulation mode is canceled.
- H'A5 is written to FKEY.
- The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When execution returns to the user procedure program, the SCO bit is cleared to 0. Therefore, the SCO bit cannot be confirmed to be 1 in the user procedure program.
The download result can be confirmed only by the return value of the DPFR parameter. Before the SCO bit is set to 1, incorrect decision must be prevented by setting the DPFR parameter, that is one byte of the start address of the on-chip RAM area specified by FTDAR, to a value other than the return value (H'FF).
When download is executed, particular interrupt processing, which is accompanied by the bank switch as described below, is performed as an internal microcomputer processing, so VBR need to be set to H'84000000. Four NOP instructions are executed immediately after the instructions that set the SCO bit to 1.

- The user MAT space is switched to the on-chip program storage area.

- After the selection condition of the download program and the address set in FTDAR are checked, the transfer processing is executed starting to the on-chip RAM address specified by FTDAR.
- The SCO bits in FCCS, FPCS, and FECS are cleared to 0.
- The return value is set to the DPFR parameter.
- After the on-chip program storage area is returned to the user MAT space, execution returns to the user procedure program.

After download is completed and the user procedure program is running, the VBR setting can be changed.

The notes on download are as follows.

In the download processing, the values of the general registers of the CPU are retained.

During the download processing, interrupts must not be generated. For details on the relationship between download and interrupts, see section 22.8.2, Interrupts during Programming/Erasing.

Since a stack area of maximum 128 bytes is used, an area of at least 128 bytes must be saved before setting the SCO bit to 1.

If flash memory is accessed by the DTC during downloading, operation cannot be guaranteed. Therefore, access by the DTC must not be executed.

(2.4) FKEY is cleared to H'00 for protection.

(2.5) The value of the DPFR parameter must be checked to confirm the download result.

A recommended procedure for confirming the download result is shown below.

- Check the value of the DPFR parameter (one byte of start address of the download destination specified by FTDAR). If the value is H'00, download has been performed normally. If the value is not H'00, the source that caused download to fail can be investigated by the description below.
- If the value of the DPFR parameter is the same as before downloading (e.g. H'FF), the address setting of the download destination in FTDAR may be abnormal. In this case, confirm the setting of the TDER bit (bit 7) in FTDAR.
- If the value of the DPFR parameter is different from before downloading, check the SS bit (bit 2) and the FK bit (bit 1) in the DPFR parameter to ensure that the download program selection and FKEY register setting were normal, respectively.

(2.6) The operating frequency is set to the FPEFEQ parameter and the user branch destination is set to the FUBRA parameter for initialization.

- The current frequency of the CPU clock is set to the FPEFEQ parameter (general register R4). For the settable range of the FPEFEQ parameter, see section 26.3.1, Clock Timing.
  When the frequency is set out of this range, an error is returned to the FPFR parameter of the initialization program and initialization is not performed. For details on the frequency setting, see the description in section 22.4.3 (2.1), Flash programming/erasing frequency parameter (FPEFEQ: general register R4 of CPU).
- The start address in the user branch destination is set to the (FUBRA: CPU general register R5) parameter.
  When the user branch processing is not required, 0 must be set to FUBRA.
  When the user branch is executed, the branch destination is executed in flash memory other than the one that is to be programmed. The area of the on-chip program that is downloaded cannot be set.
  The program processing must be returned from the user branch processing by the RTS instruction.
  See the description in section 22.4.3 (2.2), Flash user branch address setting parameter (FUBRA: general register R5 of CPU).

(2.7) Initialization

When a programming program is downloaded, the initialization program is also downloaded to on-chip RAM. There is an entry point of the initialization program in the area from (download start address set by FTDAR) + 32 bytes. The subroutine is called and initialization is executed by using the following steps.

```
MOV.L  #DLTOP+32,R1          ; Set entry address to R1
JSR    @R1                   ; Call initialization routine
NOP
```

- The general registers other than R0 are saved in the initialization program.
- R0 is a return value of the FPFR parameter.
- Since the stack area is used in the initialization program, a stack area of maximum 128 bytes must be reserved in RAM.
- Interrupts can be accepted during the execution of the initialization program. However, the program storage area and stack area in on-chip RAM and register values must not be destroyed.

(2.8) The return value of the initialization program, FPFR (general register R0) is checked.

(2.9) FKEY must be set to H'5A and the user MAT must be prepared for programming.

(2.10) The parameter which is required for programming is set.

The start address of the programming destination of the user MAT (FMPAR) is set to general register R5. The start address of the program data storage area (FMPDR) is set to general register R4.

- FMPAR setting

  FMPAR specifies the programming destination start address. When an address other than one in the user MAT area is specified, even if the programming program is executed, programming is not executed and an error is returned to the return value parameter FPFR. Since the unit is 128 bytes, the lower eight bits (MOA7 to MOA0) must be in the 128-byte boundary of H'00 or H'80.

- FMPDR setting

  If the storage destination of the program data is flash memory, even when the program execution routine is executed, programming is not executed and an error is returned to the FPFR parameter. In this case, the program data must be transferred to on-chip RAM and then programming must be executed.

(2.11) Programming

There is an entry point of the programming program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and programming is executed by using the following steps.

```
MOV.L  #DLTOP+16,R1        ; Set entry address to R1
JSR    @R1                 ; Call programming routine
NOP
```

— The general registers other than R0 are saved in the programming program.

— R0 is a return value of the FPFR parameter.

— Since the stack area is used in the programming program, a stack area of maximum 128 bytes must be reserved in RAM.

(2.12) The return value in the programming program, FPFR (general register R0) is checked.

(2.13) Determine whether programming of the necessary data has finished.

If more than 128 bytes of data are to be programmed, specify FMPAR and FMPDR in 128-byte units, and repeat steps (2.10) to (2.13). Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address which has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.

(2.14) After programming finishes, clear FKEY and specify software protection.

If this LSI is restarted by a power-on reset immediately after user MAT programming has finished, secure a reset period (period of $\overline{RES}$ = 0) that is at least as long as the normal 100 μs.

(3) Erasing Procedure in User Program Mode

The procedures for download, initialization, and erasing are shown in figure 22.12.

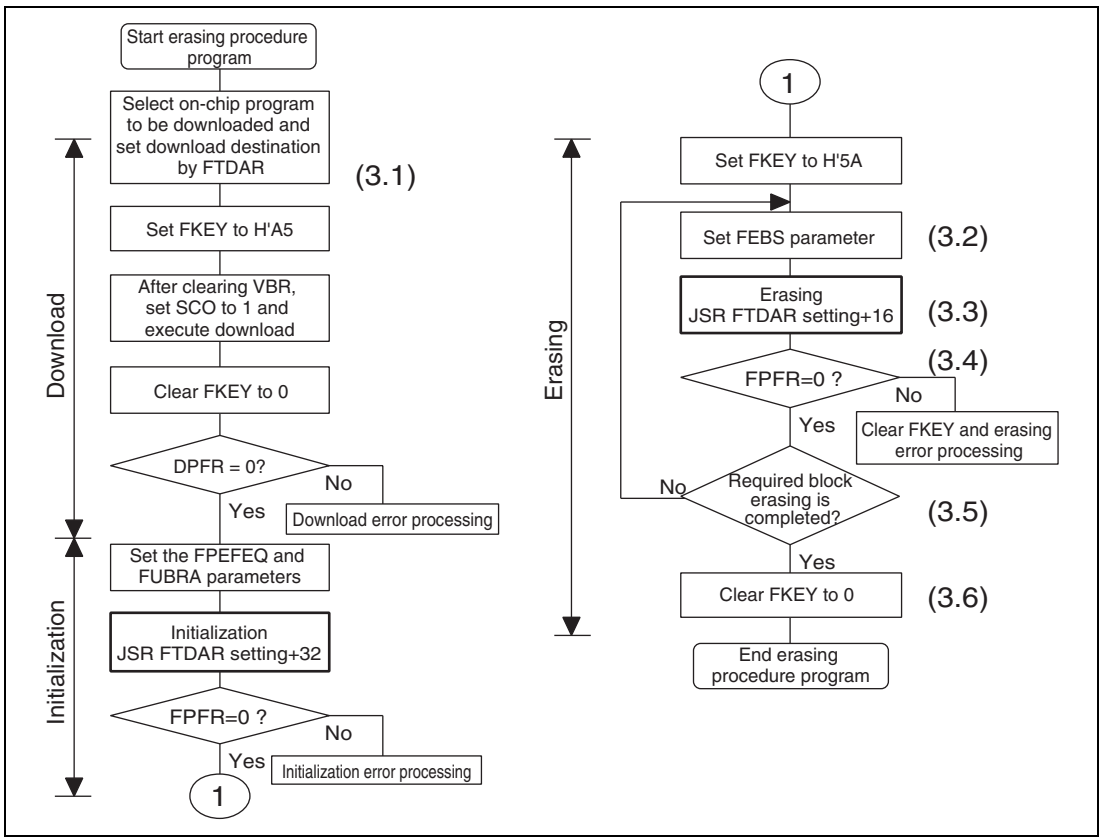



**Figure 22.12 Erasing Procedure**

The details of the erasing procedure are described below. The procedure program must be executed in an area other than the user MAT to be erased. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in on-chip RAM. Specify 1/4 (initial value) as the frequency division ratios of an internal clock (If), a bus clock (Bf), and a peripheral clock (Pf) through the frequency control register (FRQCR).

After the programming/erasing program has been downloaded and the SCO bit is cleared to 0, the setting of the frequency control register (FRQCR) can be changed to the desired value.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 22.9.2, Areas for Storage of the Procedural Program and Data for Programming.

For the downloaded on-chip program area, see the RAM map for programming/erasing in figure 22.10.

A single divided block is erased by one erasing processing. For block divisions, see figure 22.4. To erase two or more blocks, update the erase block number and perform the erasing processing for each block.

(3.1) Select the on-chip program to be downloaded

Set the EPVB bit in FECS to 1.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPFR parameter.

Specify the start address of the download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, are the same as those in the programming procedure. For details, see the description in section 22.5.2 (2), Programming Procedure in User Program Mode.

(3.2) Set the FEBS parameter necessary for erasure

Set the erase block number of the user MAT in the flash erase block select parameter (FEBS: general register R4). If a value other than an erase block number of the user MAT is set, no block is erased even though the erasing program is executed, and an error is returned to the return value parameter FPFR.

(3.3) Erasure

Similar to as in programming, there is an entry point of the erasing program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and erasing is executed by using the following steps.

```
MOV.L  #DLTOP+16,R1        ; Set entry address to R1
JSR    @R1                 ; Call erasing routine
NOP
```

— The general registers other than R0 are saved in the erasing program.

— R0 is a return value of the FPFR parameter.

— Since the stack area is used in the erasing program, a stack area of maximum 128 bytes must be reserved in RAM.

(3.4) The return value in the erasing program, FPFR (general register R0) is checked.

(3.5) Determine whether erasure of the necessary blocks has finished.

If more than one block is to be erased, update the FEBS parameter and repeat steps (3.2) to (3.5). Blocks that have already been erased can be erased again.

(3.6) After erasure finishes, clear FKEY and specify software protection.

If this LSI is restarted by a power-on reset immediately after user MAT erasing has finished, secure a reset period (period of $\overline{RES}$ = 0) that is at least as long as the normal 100 μs.

(4) Erasing and Programming Procedure in User Program Mode

By changing the on-chip RAM address of the download destination in FTDAR, the erasing program and programming program can be downloaded to separate on-chip RAM areas.

Figure 22.13 shows an example of repetitively executing RAM emulation, erasing, and programming.

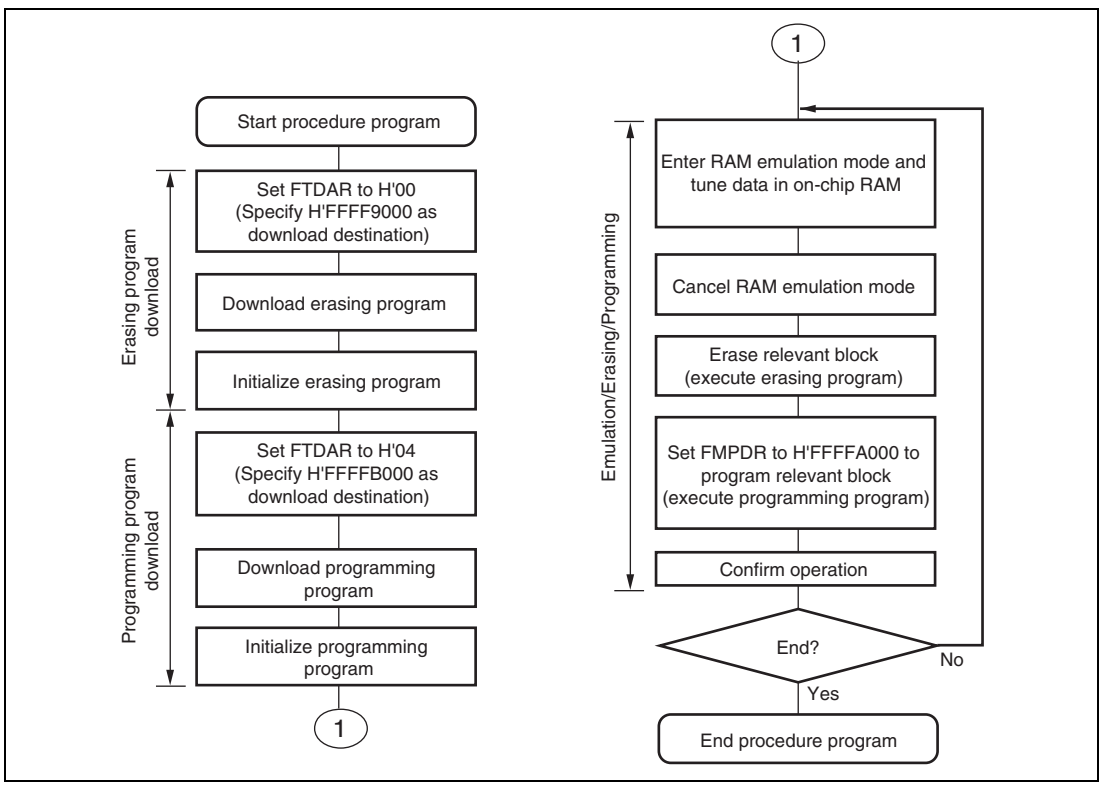



**Figure 22.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Programming (Overview)**

In the above example, the erasing program and programming program are downloaded to areas excluding addresses (H'FFFFA000 to H'FFFFAFFF) to execute RAM emulation.

Download and initialization are performed only once at the beginning.

In this kind of operation, note the following:

- Be careful not to destroy on-chip RAM with overlapped settings.

  In addition to the RAM emulation area, erasing program area, and programming program area, areas for the user procedure programs, work area, and stack area are reserved in on-chip RAM. Do not make settings that will overwrite data in these areas.
- Be sure to initialize both the erasing program and programming program.

  Initialization by setting the FPEFEQ and FUBRA parameters must be performed for both the erasing program and the programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes (H'FFFF9020 in this example) and (download start address for programming program) + 32 bytes (H'FFFFB020 in this example).

### 22.5.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings than those in user program mode or boot mode. User boot mode is a user-arbitrary boot mode, unlike boot mode that uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of the user boot MAT is only enabled in boot mode or programmer mode.

(1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 22.1.

When the reset start is executed in user boot mode, the check routine for flash-memory related registers runs. The RAM area about 1.2 Kbytes from H'FFFF9800 and 4 bytes from H'FFFFAFFC (a stack area) is used by the routine. While the check routine is running, NMI and all other interrupts cannot be accepted. This period is 100 μs while operating at an internal frequency of 40 MHz.

Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, H'AA is set to the flash MAT select register (FMATS) because the execution MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes. Figure 22.14 shows the procedure for programming the user MAT in user boot mode.

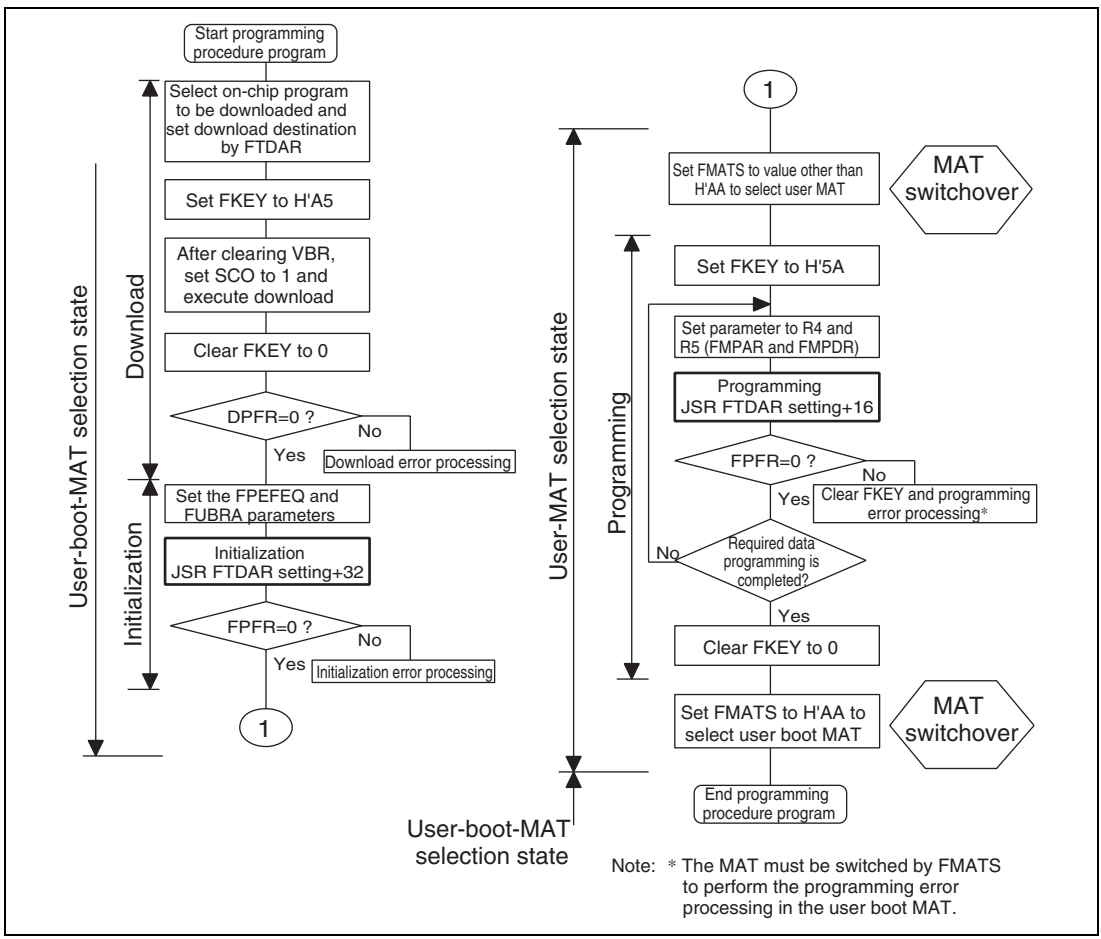



**Figure 22.14 Procedure for Programming User MAT in User Boot Mode**

The difference between the programming procedures in user program mode and user boot mode is whether the MAT is switched or not as shown in figure 22.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the user MAT hidden in the background. The user MAT and user boot MAT are switched only while the user MAT is being programmed. Because the user boot MAT is hidden while the user MAT is being programmed, the procedure program must be located in an area other than flash memory. After programming finishes, switch the MATs again to return to the first state.

MAT switchover is enabled by writing a specific value to FMATS. However note that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completely finished, and if an interrupt occurs, from which MAT the interrupt vector is read from is undetermined. Perform MAT switching in accordance with the description in section 22.8.1, Switching between User MAT and User Boot MAT.

Except for MAT switching, the programming procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 22.9.2, Areas for Storage of the Procedural Program and Data for Programming.

(3) User MAT Erasing in User Boot Mode

For erasing the user MAT in user boot mode, additional processing made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after erasing completes.

Figure 22.15 shows the procedure for erasing the user MAT in user boot mode.

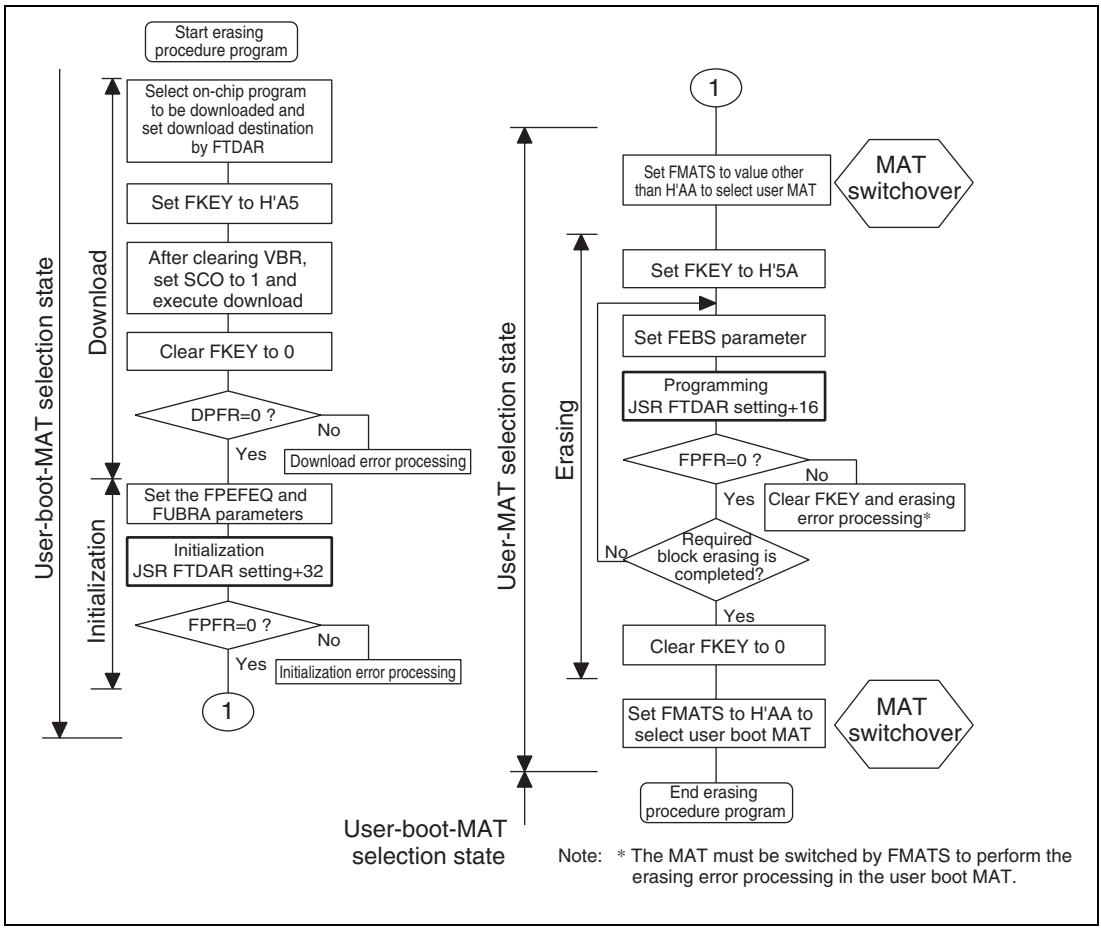



**Figure 22.15 Procedure for Erasing User MAT in User Boot Mode**

The difference between the erasing procedures in user program mode and user boot mode depends on whether the MAT is switched or not as shown in figure 22.15.

MAT switching is enabled by writing a specific value to FMATS. However note that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed finished, and if an interrupt occurs, from which MAT the interrupt vector is read from is undetermined. Perform MAT switching in accordance with the description in section 22.8.1, Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 22.9.2, Areas for Storage of the Procedural Program and Data for Programming.

## 22.6 Protection

There are three kinds of flash memory program/erase protection: hardware, software, and error protection.

### 22.6.1 Hardware Protection

Programming and erasing of flash memory is forcibly disabled or suspended by hardware protection. In this state, the downloading of an on-chip program and initialization of the flash memory are possible. However, an activated program for programming or erasure cannot program or erase locations in a user MAT, and the error in programming/erasing is reported in the FPFR parameter.

**Table 22.9 Hardware Protection**

| | | Function to be Protected | |
|---|---|---|---|
| **Item** | **Description** | **Download** | **Programming/ Erasure** |
| FWE-pin protection | The input of a low-level signal on the FWE pin clears the FWE bit of FCCS and the LSI enters a programming/erasing-protected state. | — | √ |
| Reset/standby protection | • A power-on reset (including a power-on reset by the WDT) and entry to standby mode initializes the programming/erasing interface registers and the LSI enters a programming/erasing-protected state.<br>• Resetting by means of the $\overline{\text{RES}}$ pin after power is initially supplied will not make the LSI enter the reset state unless the $\overline{\text{RES}}$ pin is held low until oscillation has stabilized. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width that is specified in the section on AC characteristics. If the LSI is reset during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again. | √ | √ |

### 22.6.2 Software Protection

Software protection is set up in any of three ways: by disabling the downloading of on-chip programs for programming and erasing, by means of a key code, and by the RAM emulation register (RAMER).

**Table 22.10 Software Protection**

| | | Function to be Protected | |
|---|---|---|---|
| **Item** | **Description** | **Download** | **Programming/ Erasure** |
| Protection by the SCO bit | Clearing the SCO bit in FCCS disables downloading of the programming/erasing program, thus making the LSI enter a programming/erasing-protected state. | √ | √ |
| Protection by FKEY | Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and for programming/erasing. | √ | √ |
| Emulation protection | Setting the RAMS bit in RAMER to 1 makes the LSI enter a programming/ erasing-protected state. | √ | √ |

### 22.6.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occurs, in the form of the microcomputer getting out of control during programming/erasing of the flash memory or operations that are not in accordance with the established procedures for programming/erasing. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, the FLER bit in FCCS is set to 1 and the LSI enters the error protection state, thus aborting programming or erasure.

The FLER bit is set to 1 in the following conditions:

- When the relevant bank area of flash memory is read during programming/erasing (including a vector read or an instruction fetch)
- When a SLEEP instruction (including software standby mode) is executed during programming/erasing

Error protection is cancelled (FLER bit is cleared) only by a power-on reset.

Note that the reset signal should only be released after providing a reset input over a period longer than the normal 100 µs. Since high voltages are applied during programming/erasing of the flash memory, some voltage may still remain even after the error protection state has been entered. For this reason, it is necessary to reduce the risk of damage to the flash memory by extending the reset period so that the charge is released.

The state-transition diagram in figure 22.16 shows transitions to and from the error protection state.

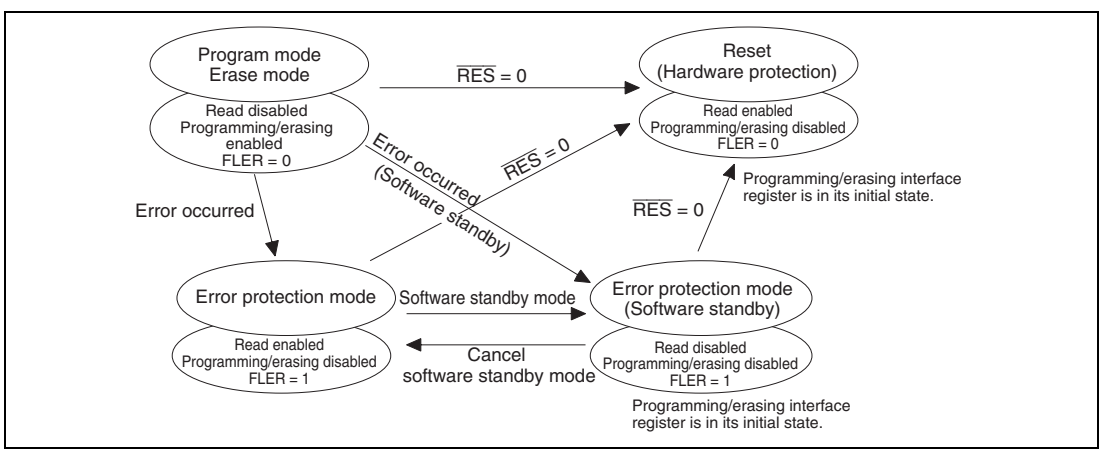



**Figure 22.16 Transitions to and from Error Protection State**

## 22.7 Flash Memory Emulation in RAM

To provide real-time emulation in RAM of data that is to be written to the flash memory, a part of the RAM can be overlaid on an area of flash memory (user MAT) that has been specified by the RAM emulation register (RAMER). After the RAMER setting is made, the RAM is accessible in both the user MAT area and as the RAM area that has been overlaid on the user MAT area. Such emulation is possible in user mode and user program mode.

Figure 22.17 shows an example of the emulation of realtime programming of the user MAT area.

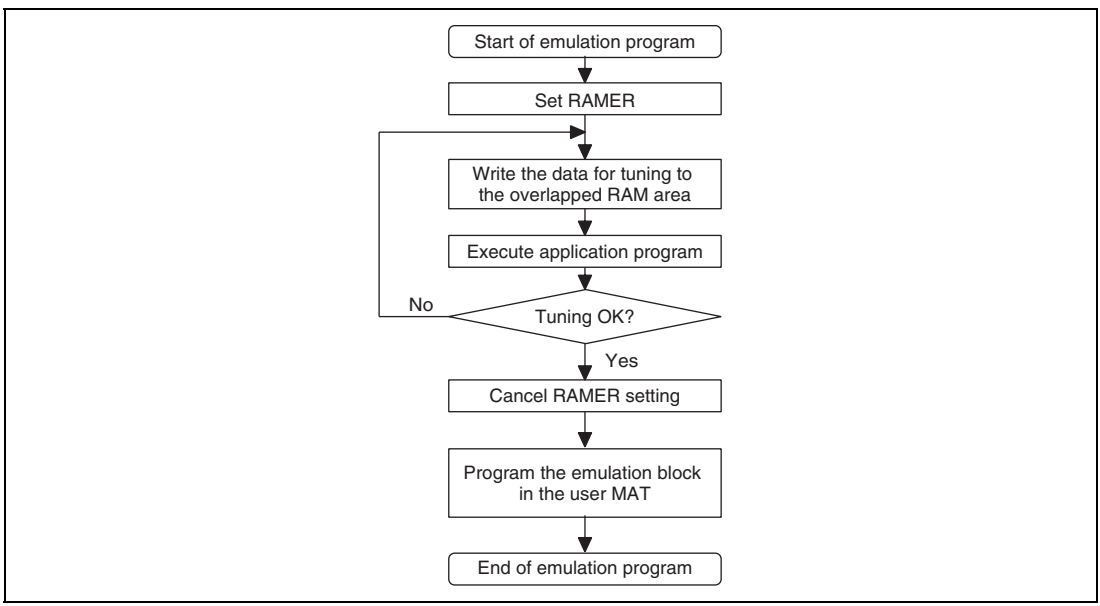



**Figure 22.17 Emulation of Flash Memory in RAM**

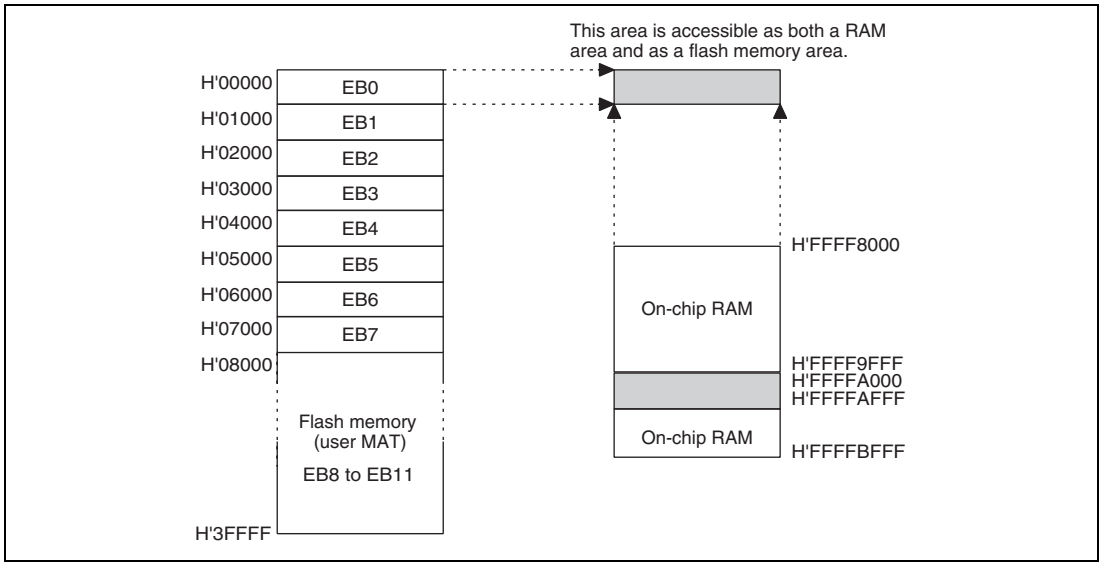



**Figure 22.18 Example of Overlapped RAM Operation (256-Kbyte Flash Memory Version)**

Figure 22.18 shows an example of an overlap on block area EB0 of the flash memory.

Emulation is possible for a single area selected from among the eight areas, from EB0 to EB7, of the user MAT. The area is selected by the setting of the RAM2 to RAM0 bits in RAMER.

1. To overlap a part of the RAM on area EB0, to allow realtime programming of the data for this area, set the RAMS bit in RAMER to 1, and each of the RAM2 to RAM0 bits to 0.
2. Realtime programming is carried out using the overlaid area of RAM.

In programming or erasing the user MAT, it is necessary to run a program that implements a series of procedural steps, including the downloading of an on-chip program. In this process, set the download area with FTDAR so that the overlaid RAM area and the area where the on-chip program is to be downloaded do not overlap.

Figure 22.19 shows an example of programming data that has been emulated to the EB0 area in the user MAT.

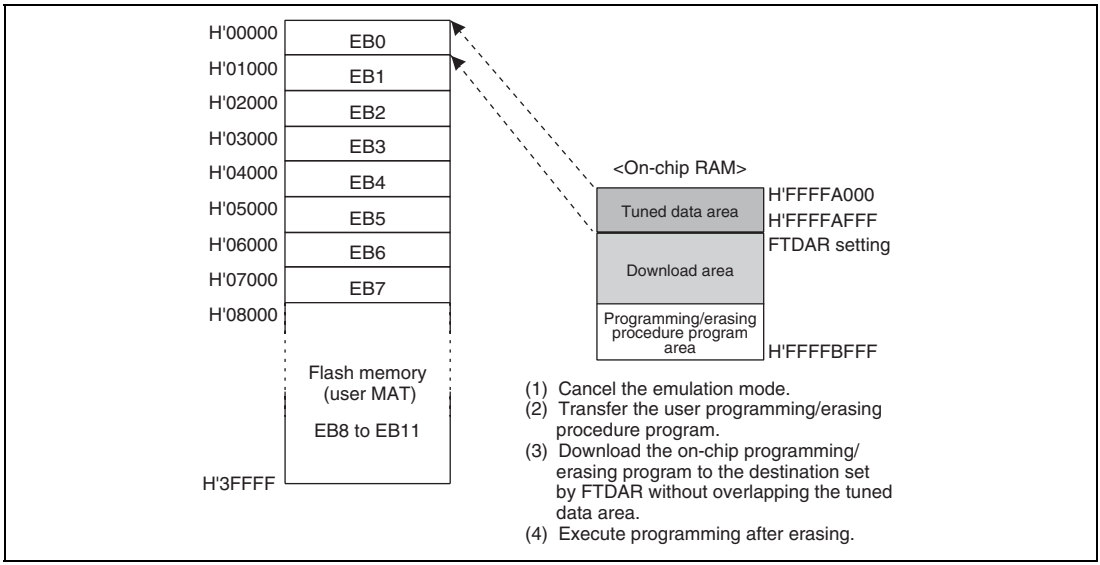



**Figure 22.19 Programming of Tuned Data (256-Kbyte Flash Memory Version)**

1. After the data to be programmed has fixed values, clear the RAMS bit to 0 to cancel the overlap of RAM. Emulation mode is canceled and emulation protection is also cleared.
2. Transfer the user programming/erasing procedure program to RAM.
3. Run the programming/erasing procedure program in RAM and download the on-chip programming/erasing program.
   Specify the download start address with FTDAR so that the tuned data area does not overlap with the download area.
4. When the EB0 area of the user MAT has not been erased, erasing must be performed before programming. Set the parameters FMPAR and FMPDR so that the tuned data is designated, and execute programming.

Note: Setting the RAMS bit to 1 puts all the blocks in flash memory in the programming/erasing-protected state regardless of the values of the RAM2 to RAM0 bits (emulation protection). Clear the RAMS bit to 0 before actual programming or erasure. Though RAM emulation can also be carried out with the user boot MAT selected, the user boot MAT can be erased or programmed only in boot mode or programmer mode.

## 22.8 Usage Notes

### 22.8.1 Switching between User MAT and User Boot MAT

It is possible to switch between the user MAT and user boot MAT. However, the following procedure is required because these MATs are allocated to address 0.

(Switching to the user boot MAT disables programming and erasing. Programming of the user boot MAT must take place in boot mode or programmer mode.)

1. MAT switching by FMATS should always be executed from the on-chip RAM. The SH microcomputer prefetches execution instructions. Therefore, a switchover during program execution in the user MAT causes an instruction code in the user MAT to be prefetched or an instruction in the newly selected user boot MAT to be prefetched, thus resulting in unstable operation.
2. To ensure that the MAT that has been switched to is accessible, execute four NOP instructions in on-chip RAM immediately after writing to FMATS of on-chip RAM (this prevents access to the flash memory during MAT switching).
3. If an interrupt occurs during switching, there is no guarantee of which memory MAT is being accessed.
   Always mask the maskable interrupts before switching MATs. In addition, configuring the system so that NMI interrupts do not occur during MAT switching is recommended.
4. After the MATs have been switched, take care because the interrupt vector table will also have been switched.
   If the same interrupt processings are to be executed before and after MAT switching or interrupt requests cannot be disabled, transfer the interrupt processing routine to on-chip RAM, and use the VBR setting to place the interrupt vector table in on chip RAM. In this case, make sure the VBR setting change does not conflict with the interrupt occurrence.
5. Memory sizes of the user MAT and user boot MAT are different. When accessing the user boot MAT, do not access addresses exceeding the 12-Kbyte memory space. If access goes beyond the 12-Kbyte space, the values read are undefined.

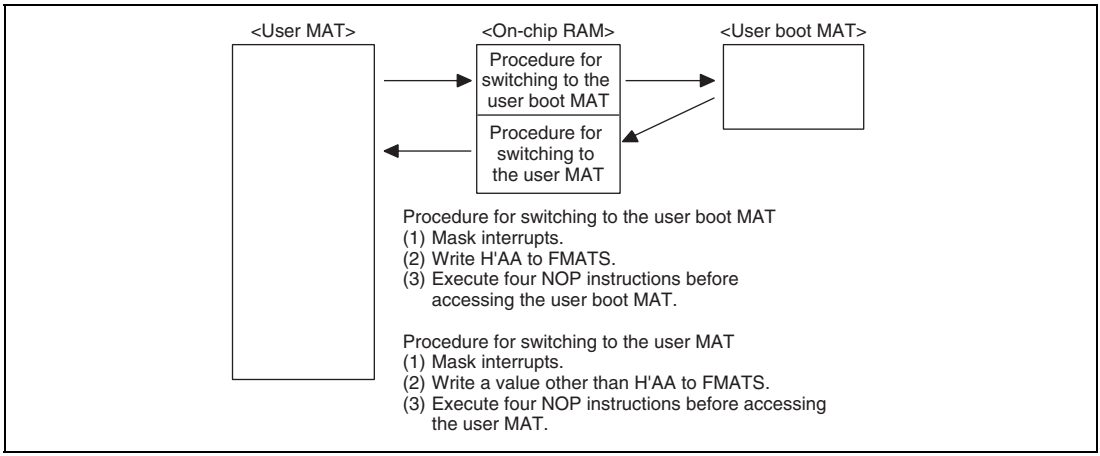



**Figure 22.20 Switching between User MAT and User Boot MAT**

### 22.8.2 Interrupts during Programming/Erasing

(1) Download of On-Chip Program

(1.1) VBR setting change

Before downloading the on-chip program, VBR must be set to H'84000000. If VBR is set to a value other than H'84000000, the interrupt vector table is placed in the user MAT (FMATS is not H'AA) or the user boot MAT (FMATS is H'AA) on setting H'84000000 to VBR.

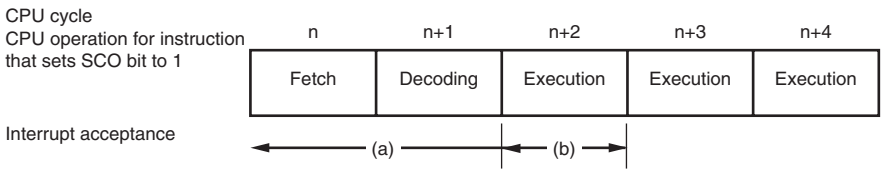
When VBR setting change conflicts with interrupt occurrence, whether the vector table before or after VBR is changed is referenced may cause an error.

Therefore, for cases where VBR setting change may conflict with interrupt occurrence, prepare a vector table to be referenced when VBR is H'00000000 (initial value) at the start of the user MAT or user boot MAT.

(1.2) SCO download request and interrupt request

Download of the on-chip programming/erasing program that is initiated by setting the SCO bit in FCCS to 1 generates a particular interrupt processing accompanied by MAT switchover. Operation when the SCO download request and interrupt request conflicts is described below.

1. Contention between SCO download request and interrupt request
   Figure 22.21 shows the timing of contention between execution of the instruction that sets the SCO bit in FCCS to 1 and interrupt acceptance.




(a) When the interrupt is accepted at the (n + 1) cycle or before
After the interrupt processing completes, the SCO bit is set to 1 and download is executed.

(b) When the interrupt is accepted at the (n + 2) cycle or later
The interrupt will conflicts with the SCO download request. Ensure that no interrupt is generated.

**Figure 22.21 Timing of Contention between SCO Download Request and Interrupt Request**

2. Generation of interrupt requests during downloading
   Ensure that interrupts are not generated during downloading that is initiated by the SCO bit.

(2) Interrupts during programming/erasing

Though an interrupt processing can be executed at realtime during programming/erasing of the downloaded on-chip program, the following limitations and notes are applied.

1. When flash memory is being programmed or erased, both the user MAT and user boot MAT cannot be accessed. Prepare the interrupt vector table and interrupt processing routine in on-chip RAM or external memory. Make sure the flash memory being programmed or erased is not accessed by the interrupt processing routine. If flash memory is read, the read values are not guaranteed. If the relevant bank in flash memory that is being programmed or erased is accessed, the error protection state is entered, and programming or erasing is aborted. If a bank other than the relevant bank is accessed, the error protection state is not entered but the read values are not guaranteed.
2. Do not rewrite the program data specified by the FMPDR parameter. If new program data is to provided by the interrupt processing, temporarily save the new program data in another area. After confirming the completion of programming, save the new program data in the area specified by FMPDR or change the setting in FMPDR to indicated the other area in which the new program data was temporarily saved.
3. Make sure the interrupt processing routine does not rewrite the contents of the flash-memory related registers or data in the downloaded on-chip program area. During the interrupt processing, do not simultaneously perform RAM emulation, download of the on-chip program by an SCO request, or programming/erasing.
4. At the beginning of the interrupt processing routine, save the CPU register contents. Before returning from the interrupt processing, write the saved contents in the CPU registers again.
5. When a transition is made to sleep mode or software standby mode in the interrupt processing routine, the error protection state is entered and programming/erasing is aborted.
   If a transition is made to the reset state, the reset signal should only be released after providing a reset input over a period longer than the normal 100 μs to reduce the damage to flash memory.

### 22.8.3 Other Notes

1. Download time of on-chip program

   The programming program that includes the initialization routine and the erasing program that includes the initialization routine are each 3 Kbytes or less. Accordingly, when the CPU clock frequency is 20 MHz, the download for each program takes approximately 10 ms at maximum.

2. User branch processing intervals

   The intervals for executing the user branch processing differs in programming and erasing. The processing phase also differs. Table 22.11 lists the maximum intervals for initiating the user branch processing when the CPU clock frequency is 80 MHz.

**Table 22.11 Initiation Intervals of User Branch Processing**

| Processing Name | Maximum Interval |
|---|---|
| Programming | Approximately 2 ms |
| Erasing | Approximately 15 ms |

However, when operation is done with CPU clock of 80 MHz, maximum values of the time until first user branch processing are as shown in table 22.12.

**Table 22.12 Initial User Branch Processing Time**

| Processing Name | Max. |
|---|---|
| Programming | Approximately 2 ms |
| Erasing | Approximately 15 ms |

3. Write to flash-memory related registers by DTC

   While an instruction in on-chip RAM is being executed, the DTC can write to the SCO bit in FCCS that is used for a download request or FMATS that is used for MAT switching. Make sure that these registers are not accidentally written to, otherwise an on-chip program may be downloaded and destroy RAM or a MAT switchover may occur and the CPU get out of control.

4. State in which interrupts are ignored

   In the following modes or period, interrupt requests are ignored; they are not executed and the interrupt sources are not retained.

   — Boot mode

   — Programmer mode

5. Note on programming the product having a 128-Kbyte user MAT

   If an attempt is made to program the product having a 128-Kbyte user MAT with more than 128 Kbytes, data programmed after the first 128 Kbytes are not guaranteed.

6. Compatibility with programming/erasing program of conventional F-ZTAT SH microcomputer

   A programming/erasing program for flash memory used in the conventional F-ZTAT SH microcomputer which does not support download of the on-chip program by a SCO transfer request cannot run in this LSI.
   Be sure to download the on-chip program to execute programming/erasing of flash memory in this LSI.

7. Monitoring runaway by WDT

   Unlike the conventional F-ZTAT SH microcomputer, no countermeasures are available for a runaway by WDT during programming/erasing by the downloaded on-chip program.
   Prepare countermeasures (e.g. use of the user branch routine and periodic timer interrupts) for WDT while taking the programming/erasing time into consideration as required.

## 22.9 Supplementary Information

### 22.9.1 Specifications of the Standard Serial Communications Interface in Boot Mode

The boot program activated in boot mode communicates with the host via the on-chip SCI of the LSI. The specifications of the serial communications interface between the host and the boot program are described below.

- States of the boot program

The boot program has three states.

1. Bit-rate matching state

   In this state, the boot program adjusts the bit rate to match that of the host. When the chip starts up in boot mode, the boot program is activated and enters the bit-rate matching state, in which it receives commands from the host and adjusts the bit rate accordingly. After bit-rate matching is complete, the boot program proceeds to the inquiry-and-selection state.
2. Inquiry-and-selection state

   In this state, the boot program responds to inquiry commands from the host. The device, clock mode, and bit rate are selected in this state. After making these selections, the boot program enters the programming/erasure state in response to the transition-to-programming/erasure state command. The boot program transfers the erasure program to RAM and executes erasure of the user MAT and user boot MAT before it enters the programming/erasure state.
3. Programming/erasure state

   In this state, programming/erasure are executed. The boot program transfers the program for programming/erasure to RAM in line with the command received from the host and executes programming/erasure. It also performs sum checking and blank checking as directed by the respective commands.

Figure 22.22 shows the flow of processing by the boot program.

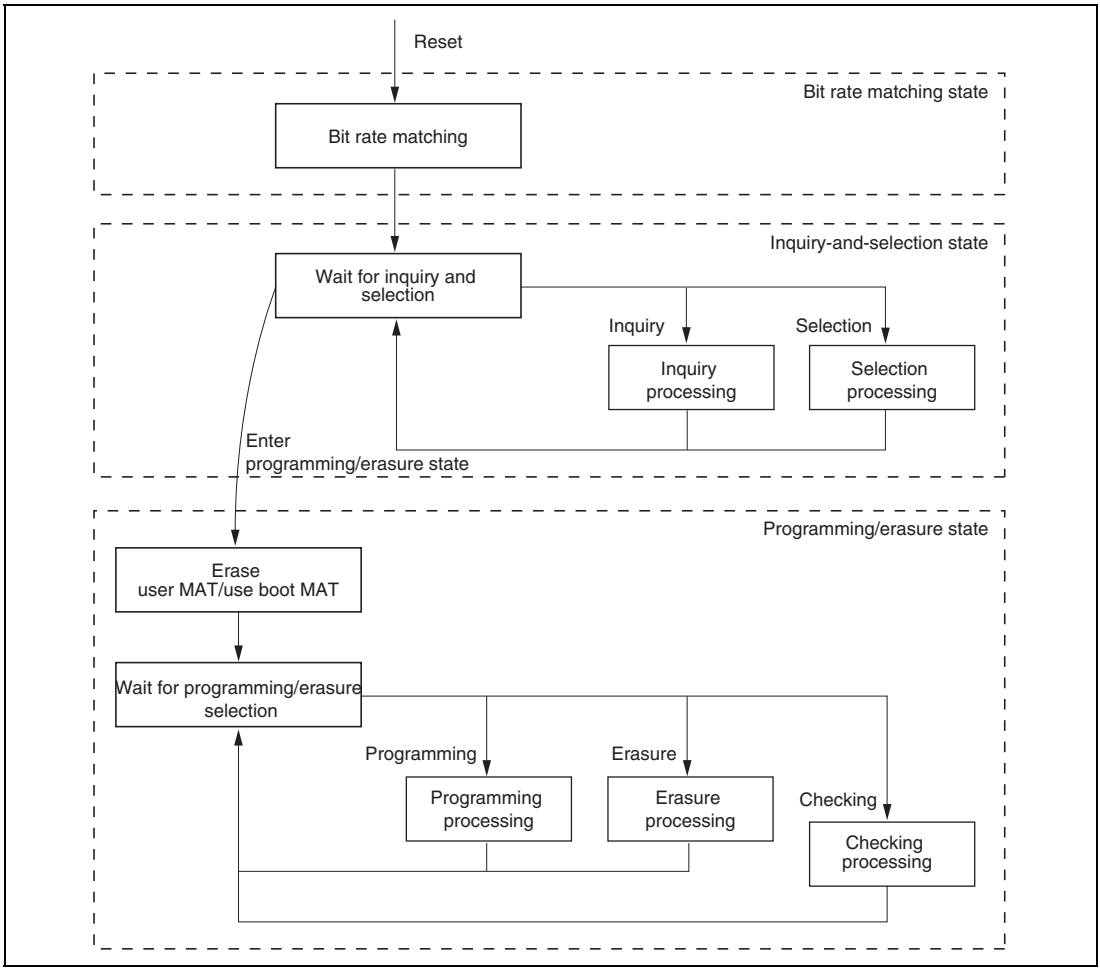



**Figure 22.22 Flow of Processing by the Boot Program**

- Bit-rate matching state

In bit-rate matching, the boot program measures the low-level intervals in a signal carrying H'00 data that is transmitted by the host, and calculates the bit rate from this. The bit rate can be changed by the new-bit-rate selection command. On completion of bit-rate matching, the boot program goes to the inquiry and selection state. The sequence of processing in bit-rate matching is shown in figure 22.23.

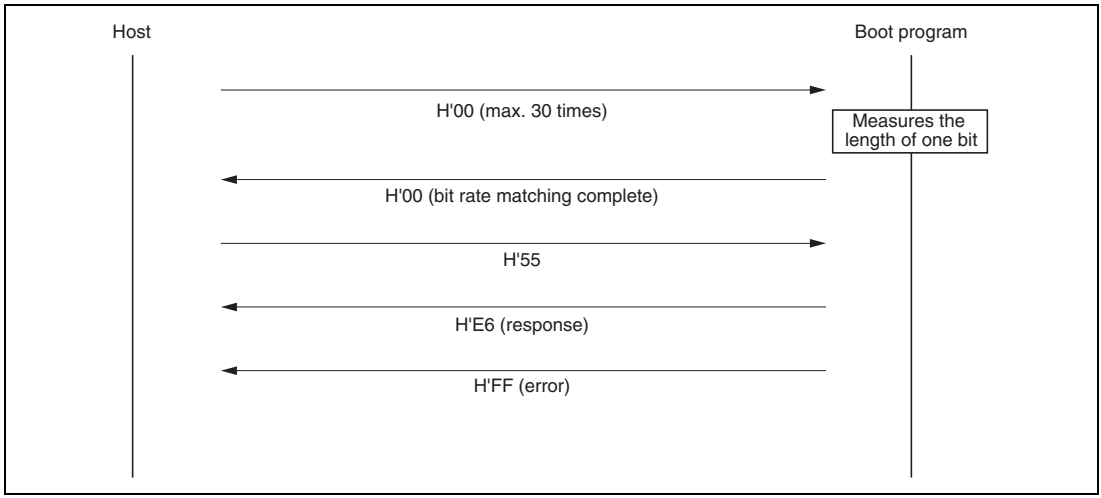



**Figure 22.23 Sequence of Bit-Rate Matching**

- Communications protocol

Formats in the communications protocol between the host and boot program after completion of the bit-rate matching are as follows.

1. One-character command or one-character response

   A command or response consisting of a single character used for an inquiry or the ACK code indicating normal completion.
2. n-character command or n-character response

   A command or response that requires n bytes of data, which is used as a selection command or response to an inquiry. The length of programming data is treated separately below.
3. Error response

   Response to a command in case of an error: two bytes, consisting of the error response and error code.
4. 128-byte programming command

   The command itself does not include data-size information. The data length is known from the response to the command for inquiring about the programming size.
5. Response to a memory reading command

   This response includes four bytes of size information.

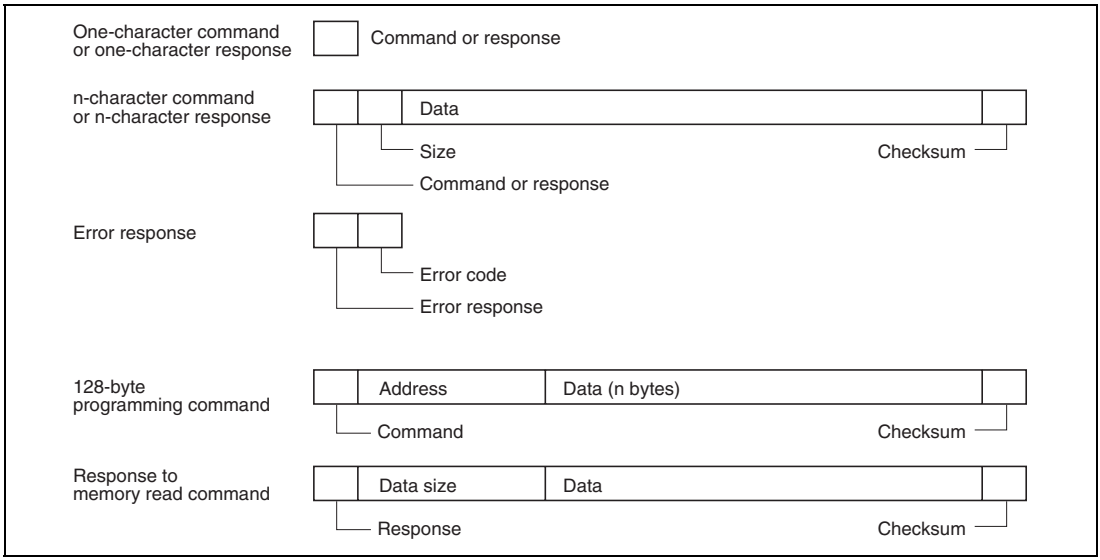



**Figure 22.24 Formats in the Communications Protocol**

— Command (1 byte): Inquiry, selection, programming, erasure, checking, etc.

— Response (1 byte): Response to an inquiry

— Size (one or two bytes): The length of data for transfer, excluding the command/response code, size, and checksum.

— Data (n bytes): Particular data for the command or response

— Checksum (1 byte): Set so that the total sum of byte values from the command code to the checksum is H'00 in the lower-order 1 byte.

— Error response (1 byte): Error response to a command

— Error code (1 byte): Indicates the type of error.

— Address (4 bytes): Address for programming

— Data (n bytes): Data to be programmed. "n" is known from the response to the command used to inquire about the programming size.

— Data size (4 bytes): Four-byte field included in the response to a memory reading command.

- **Inquiry-and-Selection State**

In this state, the boot program returns information on the flash ROM in response to inquiry commands sent from the host, and selects the device, clock mode, and bit rate in response to the respective selection commands.

The inquiry and selection commands are listed in table 22.13.

**Table 22.13 Inquiry and Selection Commands**

| Command | Command Name | Function |
|---|---|---|
| H'20 | Inquiry on supported devices | Requests the device codes and their respective boot program names. |
| H'10 | Device selection | Selects a device code. |
| H'21 | Inquiry on clock modes | Requests the number of available clock modes and their respective values. |
| H'11 | Clock-mode selection | Selects a clock mode. |
| H'22 | Inquiry on frequency multipliers | Requests the number of clock signals for which frequency multipliers and divisors are selectable, the number of multiplier and divisor settings for the respective clocks, and the values of the multipliers and divisors. |
| H'23 | Inquiry on operating frequency | Requests the minimum and maximum values for operating frequency of the main clock and peripheral clock. |
| H'24 | Inquiry on user boot MATs | Requests the number of user boot MAT areas along with their start and end addresses. |
| H'25 | Inquiry on user MATs | Requests the number of user MAT areas along with their start and end addresses. |
| H'26 | Inquiry on erasure blocks | Requests the number of erasure blocks along with their start and end addresses. |
| H'27 | Inquiry on programming size | Requests the unit of data for programming. |
| H'3F | New bit rate selection | Selects a new bit rate. |
| H'40 | Transition to programming/erasure state | On receiving this command, the boot program erases the user MAT and user boot MAT and enters the programming/erasure state. |
| H'4F | Inquiry on boot program state | Requests information on the current state of boot processing. |

The selection commands should be sent by the host in this order: device selection (H'10), clock-mode selection (H'11), new bit rate selection (H'3F). These commands are mandatory. If the same selection command is sent two or more times, the command that is sent last is effective.

All commands in the above table, except for the boot program state inquiry command (H'4F), are valid until the boot program accepts the transition-to-programming/erasure state command (H'40). That is, until the transition command is accepted, the host can continue to send commands listed in the above table until it has made the necessary inquiries and selections. The host can send the boot program state inquiry command (H'4F) even after acceptance of the transition-to-programming/erasure state command (H'40) by the boot program.

### (1) Inquiry on supported devices

In response to the inquiry on supported devices, the boot program returns the device codes of the devices it supports and the product names of their respective boot programs.

| | |
|---|---|
| Command | H'20 |

— Command H'20 (1 byte): Inquiry on supported devices

| | | | | |
|---|---|---|---|---|
| Response | H'30 | Size | No. of devices | |
| | Number of characters | Device code | | Product name |
| | … | | | |
| | SUM | | | |

— Response H'30 (1 byte): Response to the inquiry on supported devices

— Size (1 byte): The length of data for transfer excluding the command code, this field (size), and the checksum. Here, it is the total number of bytes taken up by the number of devices, number of characters, device code, and product name fields.

— Number of devices (1 byte): The number of device models supported by the boot program embedded in the microcomputer.

— Number of characters (1 byte): The number of characters in the device code and product name fields.

— Device code (4 bytes): Device code of a supported device (ASCII encoded)

— Product name (n bytes): Product code of the boot program (ASCII encoded)

— SUM (1 byte): Checksum
This is set so that the total sum of all bytes from the command code to the checksum is H'00.

### (2) Device selection

In response to the device selection command, the boot program sets the specified device as the selected device. The boot program will return the information on the selected device in response to subsequent inquiries.

| | | | | |
|---|---|---|---|---|
| Command | H'10 | Size | Device code | SUM |

— Command H'10 (1 byte): Device selection
— Size (1 byte): Number of characters in the device code (fixed at 4)
— Device code (4 bytes): A device code that was returned in response to an inquiry on supported devices (ASCII encoded)
— SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to device selection
The ACK code is returned when the specified device code matches one of the supported devices.

| | | |
|---|---|---|
| Error response | H'90 | ERROR |

— Error response H'90 (1 byte): Error response to device selection
— ERROR (1 byte): Error code
H'11: Sum-check error
H'21: Non-matching device code

### (3) Inquiry on clock modes

In response to the inquiry on clock modes, the boot program returns the number of available clock modes.

| | |
|---|---|
| Command | H'21 |

— Command H'21 (1 byte): Inquiry on clock modes

| | | | | | |
|---|---|---|---|---|---|
| Response | H'31 | Size | Mode | … | SUM |

— Response H'31 (1 byte): Response to the inquiry on clock modes
— Size (1 byte): The total length of the number of modes and mode data fields.
— Mode (1 byte): Selectable clock mode (example: H'01 denotes clock mode 1)
— SUM (1 byte): Checksum

### (4) Clock-mode selection

In response to the clock-mode selection command, the boot program sets the specified clock mode. The boot program will return the information on the selected clock mode in response to subsequent inquiries.

| | | | | |
|---|---|---|---|---|
| Command | H'11 | Size | Mode | SUM |

— Command H'11 (1 byte): Clock mode selection
— Size (1 byte): Number of characters in the clock-mode field (fixed at 1)
— Mode (1 byte): A clock mode returned in response to the inquiry on clock modes
— SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to clock mode selection
The ACK code is returned when the specified clock-mode matches one of the available clock modes.

| | | |
|---|---|---|
| Error response | H'91 | ERROR |

— Error response H'91 (1 byte): Error response to clock mode selection
— ERROR (1 byte): Error code
H'11: Sum-check error
H'21: Non-matching clock mode

### (5) Inquiry on frequency multipliers

In response to the inquiry on frequency multipliers, the boot program returns information on the settable frequency multipliers or divisors.

| | |
|---|---|
| Command | H'22 |

— Command H'22 (1 byte): Inquiry on frequency multipliers

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Response | H'32 | Size | No. of operating clocks | | | | | |
| | No. of multipliers | Multiplier | … | | | | | |
| | … | | | | | | | |
| | SUM | | | | | | | |

— Response H'32 (1 byte): Response to the inquiry on frequency multipliers

— Size (1 byte): The total length of the number of operating clocks, number of multipliers, and multiplier fields.

— Number of operating clocks (1 byte): The number of operating clocks for which multipliers can be selected
(for example, if frequency multiplier settings can be made for the frequencies of the main and peripheral operating clocks, the value should be H'02).

— Number of multipliers (1 byte): The number of multipliers selectable for the operating frequency of the main or peripheral modules

— Multiplier (1 byte):
Multiplier: Numerical value in the case of frequency multiplication (e.g. H'04 for ×4)
Divisor: Two's complement negative numerical value in the case of frequency division (e.g. H'FE [-2] for ×1/2)
As many multiplier fields are included as there are multipliers or divisors, and combinations of the number of multipliers and multiplier fields are repeated as many times as there are operating clocks.

— SUM (1 byte): Checksum

### (6) Inquiry on operating frequency

In response to the inquiry on operating frequency, the boot program returns the number of operating frequencies and the maximum and minimum values.

Command

| |
|---|
| H'23 |

— Command H'23 (1 byte): Inquiry on operating frequency

Response

| | | |
|---|---|---|
| H'33 | Size | No. of operating clocks |
| Operating freq. (min) | | Operating freq. (max) |
| … | | |
| SUM | | |

— Response H'33 (1 byte): Response to the inquiry on operating frequency

— Size (1 byte): The total length of the number of operating clocks, and maximum and minimum values of operating frequency fields.

— Number of operating clocks (1 byte): The number of operating clock frequencies required within the device.
For example, the value two indicates main and peripheral operating clock frequencies.

— Minimum value of operating frequency (2 bytes): The minimum frequency of a frequency-multiplied or -divided clock signal.
The value in this field and in the maximum value field is the frequency in MHz to two decimal places, multiplied by 100 (for example, if the frequency is 20.00 MHz, the value multiplied by 100 is 2000, so H'07D0 is returned here).

— Maximum value of operating frequency (2 bytes): The maximum frequency of a frequency-multiplied or -divided clock signal.
As many pairs of minimum/maximum values are included as there are operating clocks.

— SUM (1 byte): Checksum

### (7) Inquiry on user boot MATs

In response to the inquiry on user boot MATs, the boot program returns the number of user boot MAT areas and their addresses.

| | |
|---|---|
| Command | H'24 |

— Command H'24 (1 byte): Inquiry on user boot MAT information

| | | | | |
|---|---|---|---|---|
| Response | H'34 | Size | No. of areas | |
| | First address of the area | | | Last address of the area |
| | … | | | |
| | SUM | | | |

— Response H'34 (1 byte): Response to the inquiry on user boot MATs
— Size (1 byte): The total length of the number of areas and first and last address fields.
— Number of areas (1 byte): The number of user boot MAT areas.
H'01 is returned if the entire user boot MAT area is continuous.
— First address of the area (4 bytes)
— Last address of the area (4 bytes)
As many pairs of first and last address field are included as there are areas.
— SUM (1 byte): Checksum

### (8) Inquiry on user MATs

In response to the inquiry on user MATs, the boot program returns the number of user MAT areas and their addresses.

| | |
|---|---|
| Command | H'25 |

— Command H'25 (1 byte): Inquiry on user MAT information

| | | | | |
|---|---|---|---|---|
| Response | H'35 | Size | No. of areas | |
| | First address of the area | | | Last address of the area |
| | … | | | |
| | SUM | | | |

— Response H'35 (1 byte): Response to the inquiry on user MATs
— Size (1 byte): The total length of the number of areas and first and last address fields.
— Number of areas (1 byte): The number of user MAT areas.
H'01 is returned if the entire user MAT area is continuous.
— First address of the area (4 bytes)
— Last address of the area (4 bytes)
As many pairs of first and last address field are included as there are areas.
— SUM (1 byte): Checksum

### (9) Inquiry on erasure blocks

In response to the inquiry on erasure blocks, the boot program returns the number of erasure blocks in the user MAT and the addresses where each block starts and ends.

| | |
|---|---|
| Command | H'26 |

— Command H'26 (1 byte): Inquiry on erasure blocks

| | | | | |
|---|---|---|---|---|
| Response | H'36 | Size | No. of blocks | |
| | First address of the block | | | Last address of the block |
| | … | | | |
| | SUM | | | |

— Response H'36 (1 byte): Response to the inquiry on erasure blocks
— Size (2 bytes): The total length of the number of blocks and first and last address fields.
— Number of blocks (1 byte): The number of erasure blocks in flash memory
— First address of the block (4 bytes)
— Last address of the block (4 bytes)
As many pairs of first and last address data are included as there are blocks.
— SUM (1 byte): Checksum

### (10) Inquiry on programming size

In response to the inquiry on programming size, the boot program returns the size, in bytes, of the unit for programming.

| | | |
|---|---|---|
| Command | H'27 | |

— Command H'27 (1 byte): Inquiry on programming size

| | | | | |
|---|---|---|---|---|
| Response | H'37 | Size | Programming size | SUM |

— Response H'37 (1 byte): Response to the inquiry on programming size
— Size (1 byte): The number of characters in the programming size field (fixed at 2)
— Programming size (2 bytes): The size of the unit for programming
This is the unit for the reception of data to be programmed.
— SUM (1 byte): Checksum

### (11) New bit rate selection

In response to the new-bit-rate selection command, the boot program changes the bit rate setting to the new bit rate and, if the setting was successful, responds to the ACK sent by the host by returning another ACK at the new bit rate.

The new-bit-rate selection command should be sent after clock-mode selection.

| | | | | |
|---|---|---|---|---|
| Command | H'3F | Size | Bit rate | Input frequency |
| | No. of multipliers | Multiplier 1 | Multiplier 2 | |
| | SUM | | | |

— Command H'3F (1 byte): New bit rate selection
— Size (1 byte): The total length of the bit rate, input frequency, number of multipliers, and multiplier fields
— Bit rate (2 bytes): New bit rate
The bit rate value divided by 100 should be set here (for example, to select 19200 bps, the set H'00C0, which is 192 in decimal notation).
— Input frequency (2 bytes): The frequency of the clock signal fed to the boot program
This should be the frequency in MHz to the second decimal place, multiplied by 100 (for example, if the frequency is 28.882 MHz, the values is truncated to the second decimal place and multiplied by 100, making 2888; so H'0B48 should be set in this field).

— Number of multipliers (1 byte): The number of selectable frequency multipliers and divisors for the device.
This is normally 2, which indicates the main operating frequency and the operating frequency of the peripheral modules.

— Multiplier 1 (1 byte): Multiplier or divisor for the main operating frequency
Multiplier: Numerical value of the frequency multiplier (e.g. H'04 for ×4)
Divisor: Two’s complement negative numerical value in the case of frequency division (e.g. H'FE [-2] for ×1/2)

— Multiplier 2 (1 byte): Multiplier or divisor for the peripheral operating frequency
Multiplier: Numerical value of the frequency multiplier (e.g. H'04 for ×4)
Divisor: Two's complement negative numerical value in the case of frequency division (e.g. H'FE [-2] for ×1/2)

— SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to the new-bit-rate selection command
The ACK code is returned if the specified bit rate was selectable.

| | | |
|---|---|---|
| Error response | H'BF | ERROR |

— Error response H'BF (1 byte): Error response to new bit rate selection

— ERROR (1 byte): Error code

H'11: Sum-check error

H'24: Bit rate selection error (the specified bit rate is not selectable).

H'25: Input frequency error (the specified input frequency is not within the range from the minimum to the maximum value).

H'26: Frequency multiplier error (the specified multiplier does not match an available one).

H'27: Operating frequency error (the specified operating frequency is not within the range from the minimum to the maximum value).

The received data are checked in the following ways.

1. Input frequency

   The value of the received input frequency is checked to see if it is within the range of the minimum and maximum values of input frequency for the selected clock mode of the selected device. A value outside the range generates an input frequency error.

2. Multiplier

   The value of the received multiplier is checked to see if it matches a multiplier or divisor that is available for the selected clock mode of the selected device. A value that does not match an available ratio generates a frequency multiplier error.

3. Operating frequency

   The operating frequency is calculated from the received input frequency and the frequency multiplier or divisor. The input frequency is the frequency of the clock signal supplied to the LSI, while the operating frequency is the frequency at which the LSI is actually driven. The following formulae are used for this calculation.

   Operating frequency = input frequency × multiplier, or

   Operating frequency = input frequency / divisor

   The calculated operating frequency is checked to see if it is within the range of the minimum and maximum values of the operating frequency for the selected clock mode of the selected device. A value outside the range generates an operating frequency error.

4. Bit rate

   From the peripheral operating frequency (Pϕ) and the bit rate (B), the value (= n) of the clock select bits (CKS) in the serial mode register (SCSMR) and the value (= N) of the bit rate register (SCBRR) are calculated, after which the error in the bit rate is calculated. This error is checked to see if it is smaller than 4%. A result greater than or equal to 4% generates a bit rate selection error. The following formula is use to calculate the error.

$$\text{Error (\%)} = \left\{ \left[ \frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} \right] - 1 \right\} \times 100$$

When the new bit rate is selectable, the boot program returns an ACK code to the host and then makes the register setting to select the new bit rate. The host then sends an ACK code at the new bit rate, and the boot program responds to this with another ACK code, this time at the new bit rate.

| | |
|---|---|
| Acknowledge | H'06 |

— Acknowledge H'06 (1 byte): The ACK code sent by the host to acknowledge the new bit rate.

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): The ACK code transferred in response to acknowledgement of the new bit rate

The sequence of new bit rate selection is shown in figure 22.25.

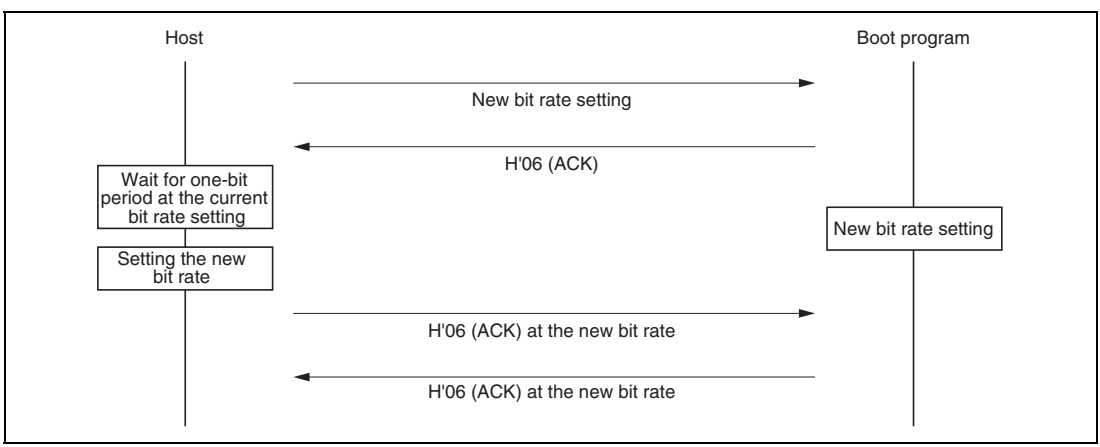



**Figure 22.25 Sequence of New Bit Rate Selection**

### (12) Transition to the programming/erasure state

In response to the transition to the programming/erasure state command, the boot program transfers the erasing program and runs it to erase any data in the user MAT and then the user boot MAT. On completion of this erasure, the boot program returns the ACK code and enters the programming/erasure state.

Before sending the programming selection command and data for programming, the host must select the device, clock mode, and new bit rate for the LSI by issuing the device selection command, clock-mode selection command, new-bit-rate selection command, and then initiate the transition to the programming/erasure state by sending the corresponding command to the boot program.

| | |
|---|---|
| Command | H'40 |

— Command H'40 (1 byte): Transition to programming/erasure state

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to the transition-to-programming/erasure state command
This is returned as ACK when erasure of the user boot MAT and user MAT has succeeded after transfer of the erasure program.

| | | |
|---|---|---|
| Error response | H'C0 | H'51 |

— Error response H'C0 (1 byte): Error response to the transition-to-programming/erasure state command

— ERROR (1 byte): Error code
H'51: Erasure error (Erasure did not succeed because of an error.)

- Command Error

Command errors are generated by undefined commands, commands sent in an incorrect order, and the inability to accept a command. For example, sending the clock-mode selection command before device selection or an inquiry command after the transition-to-programming/erasure state command generates a command error.

| | | |
|---|---|---|
| Error response | H'80 | H'xx |

— Error response H'80 (1 byte): Command error

— Command H'xx (1 byte): Received command

- Order of Commands

In the inquiry-and-selection state, commands should be sent in the following order.

1. Send the inquiry on supported devices command (H'20) to get the list of supported devices.
2. Select a device from the returned device information, and send the device selection command (H'10) to select that device.
3. Send the inquiry on clock mode command (H'21) to get the available clock modes.
4. Select a clock mode from among the returned clock modes, and send the clock-mode selection command (H'11).
5. After selection of the device and clock mode, send the commands to inquire about frequency multipliers (H'22) and operating frequencies (H'23) to get the information required to select a new bit rate.
6. Taking into account the returned information on the frequency multipliers and operating frequencies, send a new-bit-rate selection command (H'3F).
7. After the device and clock mode have been selected, get the information required for programming and erasure of the user boot MAT and user MAT by sending the commands to inquire about the user boot MAT (H'24), user MAT (H'25), erasure block (H'26), and programming size (H'27).
8. After making all necessary inquiries and the new bit rate selection, send the transition-to-programming/erasure state command (H'40) to place the boot program in the programming/erasure state.

- **Programming/Erasure State**

In this state, the boot program must select the form of programming corresponding to the programming-selection command and then write data in response to 128-byte programming commands, or perform erasure in block units in response to the erasure-selection and block-erasure commands.

The programming and erasure commands are listed in table 22.14.

**Table 22.14 Programming and Erasure Commands**

| Command | Command Name | Function |
|---|---|---|
| H'42 | Selection of user boot MAT programming | Selects transfer of the program for user boot MAT programming. |
| H'43 | Selection of user MAT programming | Selects transfer of the program for user MAT programming. |
| H'50 | 128-byte programming | Executes 128-byte programming. |
| H'48 | Erasure selection | Selects transfer of the erasure program. |
| H'58 | Block erasure | Executes erasure of the specified block. |
| H'52 | Memory read | Reads from memory. |
| H'4A | Sum checking of user boot MAT | Executes sum checking of the user boot MAT. |
| H'4B | Sum checking of user MAT | Executes sum checking of the user MAT. |
| H'4C | Blank checking of user boot MAT | Executes blank checking of the user boot MAT. |
| H'4D | Blank checking of user MAT | Executes blank checking of the user MAT. |
| H'4F | Inquiry on boot program state | Requests information on the state of boot processing. |

- Programming

Programming is performed by issuing a programming-selection command and the 128-byte programming command.

Firstly, the host issues the programming-selection command to select the MAT to be programmed. Two programming-selection commands are provided for the selection of either of the two target areas.

1. Selection of user boot MAT programming
2. Selection of user MAT programming

Next, the host issues a 128-byte programming command. 128 bytes of data for programming by the method selected by the preceding programming selection command are expected to follow the command. To program more than 128 bytes, repeatedly issue 128-byte programming commands. To terminate programming, the host should send another 128-byte programming command with the address H'FFFFFFFF. On completion of programming, the boot program waits for the next programming/erasure selection command.

To then program the other MAT, start by sending the programming select command.

The sequence of programming by programming-selection and 128-byte programming commands is shown in figure 22.26.

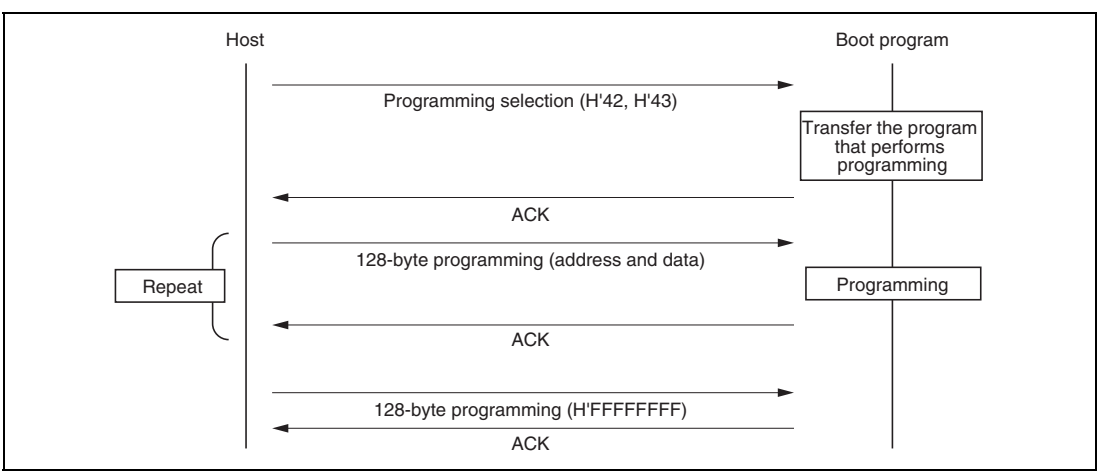



**Figure 22.26 Sequence of Programming**

### (1) Selection of user boot MAT programming

In response to the command for selecting programming of the user boot MAT, the boot program transfers the corresponding flash-writing program, i.e. the program for writing to the user boot MAT.

| | |
|---|---|
| Command | H'42 |

— Command H'42 (1 byte): Selects programming of the user boot MAT.

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to selection of user boot MAT programming
This ACK code is returned after transfer of the program that performs writing to the user boot MAT.

| | | |
|---|---|---|
| Error response | H'C2 | ERROR |

— Error response H'C2 (1 byte): Error response to selection of user boot MAT programming
— ERROR (1 byte): Error code
H'54: Error in selection processing (processing was not completed because of a transfer error)

### (2) Selection of user MAT programming

In response to the command for selecting programming of the user MAT, the boot program transfers the corresponding flash-writing program, i.e. the program for writing to the user MAT.

| | |
|---|---|
| Command | H'43 |

— Command H'43 (1 byte): Selects programming of the user MAT.

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to selection of user MAT programming
This ACK code is returned after transfer of the program that performs writing to the user MAT.

Error response

| | |
|---|---|
| H'C3 | ERROR |

— Error response H'C3 (1 byte): Error response to selection of user MAT programming
— ERROR (1 byte): Error code
H'54: Error in selection processing (processing was not completed because of a transfer error)

### (3) 128-byte programming

In response to the 128-byte programming command, the boot program executes the flash-writing program transferred in response to the command to select programming of the user boot MAT or user MAT.

Command

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| H'50 | Address for programming | | | | | | |
| Data | … | | | | | | |
| … | | | | | | | |
| SUM | | | | | | | |

— Command H'50 (1 byte): 128-byte programming
— Address for programming (4 bytes): Address where programming starts
This should be the address of a 128-byte boundary.
[Example] H'00, H01, H'00, H'00: H'00010000
— Programming data (n bytes): Data for programming
The length of the programming data is the size returned in response to the programming size inquiry command.
— SUM (1 byte): Checksum

Response

| |
|---|
| H'06 |

— Response H'06 (1 byte): Response to 128-byte programming
The ACK code is returned on completion of the requested programming.

Error response

| | |
|---|---|
| H'D0 | ERROR |

— Error response H'D0 (1 byte): Error response to 128-byte programming
— ERROR (1 byte): Error code
H'11: Sum-check error
H'2A: Address error (the address is not within the range for the selected MAT)
H'53: Programming error (programming failed because of an error in programming)

The specified address should be on a boundary corresponding to the unit of programming (programming size). For example, when programming 128 bytes of data, the lowest byte of the address should be either H'00 or H'80. When less than 128 bytes of data are to be programmed, the host should transmit the data after padding the vacant bytes with H'FF.

To terminate programming of a given MAT, send a 128-byte programming command with the address field H'FFFFFFFF. This informs the boot program that all data for the selected MAT have been sent; the boot program then waits for the next programming/erasure selection command.

| | | | |
|---|---|---|---|
| Command | H'50 | Address for programming | SUM |

— Command H'50 (1 byte): 128-byte programming
— Address for programming (4 bytes): Terminating code (H'FF, H'FF, H'FF, H'FF)
— SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to 128-byte programming
This ACK code is returned on completion of the requested programming.

| | | |
|---|---|---|
| Error response | H'D0 | ERROR |

— Error response H'D0 (1 byte): Error response to 128-byte programming
— ERROR (1 byte): Error code
H'11: Sum-check error
H'53: Programming error

- Erasure

Erasure is performed by issuing the erasure selection command and then one or more block erasure commands.

Firstly, the host sends the erasure selection command to select erasure; after that, it sends a block erasure command to actually erase a specific block. To erase multiple blocks, send further block erasure commands. To terminate erasure, the host should send a block erasure command with the block number H'FF. After this, the boot program waits for the next programming/erasure selection command.

The sequence of erasure by the erasure selection command and block erasure command is shown in figure 22.27.

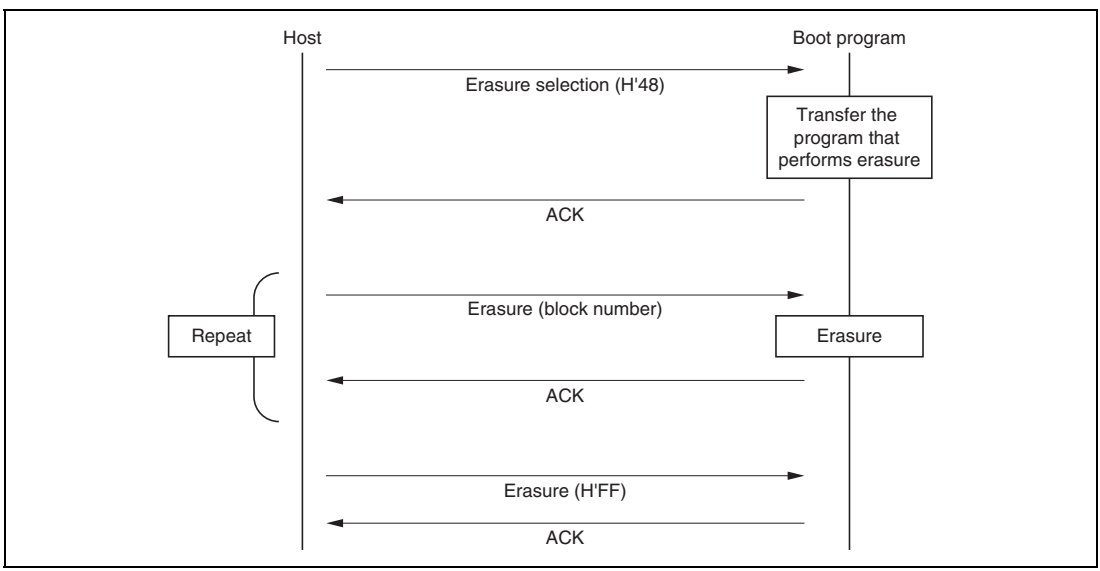



**Figure 22.27 Sequence of Erasure**

### (4) Select erasure

In response to the erasure selection command, the boot program transfers the program that performs erasure, i.e. erases data in the user MAT.

| | |
|---|---|
| Command | H'48 |

— Command H'48 (1 byte): Selects erasure.

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to selection of erasure
This ACK code is returned after transfer of the program that performs erasure.

| | | |
|---|---|---|
| Error response | H'C8 | ERROR |

— Error response H'C8 (1 byte): Error response to selection of erasure
— ERROR (1 byte): Error code
H'54: Error in selection processing (processing was not completed because of a transfer error.)

### (2) Block erasure

In response to the block erasure command, the boot program erases the data in a specified block of the user MAT.

| | | | | |
|---|---|---|---|---|
| Command | H'58 | Size | Block number | SUM |

— Command H'58 (1 byte): Erasure of a block
— Size (1 byte): The number of characters in the block number field (fixed at 1)
— Block number (1 byte): Block number of the block to be erased
— SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to the block erasure command
This ACK code is returned when the block has been erased.

| | | |
|---|---|---|
| Error response | H'D8 | ERROR |

— Error response H'D8 (1 byte): Error response to the block erasure command
— ERROR (1 byte): Error code
H'11: Sum-check error
H'29: Block number error (the specified block number is incorrect.)
H'51: Erasure error (an error occurred during erasure.)

On receiving the command with H'FF as the block number, the boot program stops erasure processing and waits for the next programming/erasure selection command.

| | | | | |
|---|---|---|---|---|
| Command | H'58 | Size | Block number | SUM |

— Command H'58 (1 byte): Erasure of a block
— Size (1 byte): The number of characters in the block number field (fixed at 1)
— Block number (1 byte): H'FF (erasure terminating code)
— SUM (1 byte): Checksum

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): ACK code to indicate response to the request for termination of erasure

To perform erasure again after having issued the command with the block number specified as H'FF, execute the process from the selection of erasure.

- Memory read

In response to the memory read command, the boot program returns the data from the specified address.

| | | | | | |
|---|---|---|---|---|---|
| Command | H'52 | Size | Area | First address for reading | |
| | Amount to read | | | | SUM |

— Command H'52 (1 byte): Memory read
— Size (1 byte): The total length of the area, address for reading, and amount to read fields (fixed value of 9)

— Area (1 byte):
H'00: User boot MAT
H'01: User MAT
An incorrect area specification will produce an address error.

— Address where reading starts (4 bytes)

— Amount to read (4 bytes): The amount of data to be read

— SUM (1 byte): Checksum

Response

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| H'52 | Amount to read | | | | | | |
| Data | … | | | | | | |
| SUM | | | | | | | |

— Response H'52 (1 byte): Response to the memory read command

— Amount to read (4 bytes): The amount to read as specified in the memory read command

— Data (n bytes): The specified amount of data read out from the specified address

— SUM (1 byte): Checksum

Error response

| | |
|---|---|
| H'D2 | ERROR |

— Error response H'D2 (1 byte): Error response to memory read command

— ERROR (1 byte): Error code
H'11: Sum-check error
H'2A: Address error (the address specified for reading is beyond the range of the MAT)
H'2B: Size error (the specified amount is greater than the size of the MAT,
the last address for reading as calculated from the specified address for the start of reading and the amount to read is beyond the MAT area, or "0" was specified as the amount to read)

- Sum checking of the user boot MAT

In response to the command for sum checking of the user boot MAT, the boot program adds all bytes of data in the user boot MAT and returns the result.

| | |
|---|---|
| Command | H'4A |

— Command H'4A (1 byte): Sum checking of the user boot MAT

| | | | | |
|---|---|---|---|---|
| Response | H'5A | Size | Checksum for the MAT | SUM |

— Response H'5A (1 byte): Response to sum checking of the user boot MAT
— Size (1 byte): The number of characters in the checksum for the MAT (fixed at 4)
— Checksum for the MAT (4 bytes): Result of checksum calculation for the user boot MAT: the total of all data in the MAT, in byte units.
— SUM (1 byte): Checksum (for the transmitted data)

- Sum checking of the user MAT

In response to the command for sum checking of the user MAT, the boot program adds all bytes of data in the user MAT and returns the result.

| | |
|---|---|
| Command | H'4B |

— Command H'4B (1 byte): Sum checking of the user MAT

| | | | | |
|---|---|---|---|---|
| Response | H'5B | Size | Checksum for the MAT | SUM |

— Response H'5B (1 byte): Response to sum checking of the user MAT
— Size (1 byte): The number of characters in the checksum for the MAT (fixed at 4)
— Checksum for the MAT (4 bytes): Result of checksum calculation for the user MAT: the total of all data in the MAT, in byte units.
— SUM (1 byte): Checksum (for the transmitted data)

- Blank checking of the user boot MAT

In response to the command for blank checking of the user boot MAT, the boot program checks to see if the whole of the user boot MAT is blank; the value returned indicates the result.

| | |
|---|---|
| Command | H'4C |

— Command H'4C (1 byte): Blank checking of the user boot MAT

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to blank checking of the user boot MAT
This ACK code is returned when the whole area is blank (all bytes are H'FF).

| | | |
|---|---|---|
| Error response | H'CC | H'52 |

— Error response H'CC (1 byte): Error response to blank checking of the user boot MAT
— Error code H'52 (1 byte): Non-erased error

- Blank checking of the user MAT

In response to the command for blank checking of the user MAT, the boot program checks to see if the whole of the user MAT is blank; the value returned indicates the result.

| | |
|---|---|
| Command | H'4D |

— Command H'4D (1 byte): Blank checking of the user boot MAT

| | |
|---|---|
| Response | H'06 |

— Response H'06 (1 byte): Response to blank checking of the user MAT
The ACK code is returned when the whole area is blank (all bytes are H'FF).

| | | |
|---|---|---|
| Error response | H'CD | H'52 |

— Error response H'CD (1 byte): Error response to blank checking of the user MAT
— Error code H'52 (1 byte): Non-erased error

- Inquiry on boot program state

In response to the command for inquiry on the state of the boot program, the boot program returns an indicator of its current state and error information. This inquiry can be made in the inquiry-and-selection state or the programming/erasure state.

| | |
|---|---|
| Command | H'4F |

— Command H'4F (1 byte): Inquiry on boot program state

| | | | | | |
|---|---|---|---|---|---|
| Response | H'5F | Size | STATUS | ERROR | SUM |

— Response H'5F (1 byte): Response to the inquiry regarding boot-program state

— Size (1 byte): The number of characters in STATUS and ERROR (fixed at 2)

— STATUS (1 byte): State of the standard boot program
See table 22.15, Status Codes.

— ERROR (1 byte): Error state (indicates whether the program is in normal operation or an error has occurred)
ERROR = 0: Normal
ERROR ≠ 0: Error
See table 22.16, Error Codes.

— SUM (1 byte): Checksum

**Table 22.15 Status Codes**

| Code | Description |
|---|---|
| H'11 | Waiting for device selection |
| H'12 | Waiting for clock-mode selection |
| H'13 | Waiting for bit-rate selection |
| H'1F | Waiting for transition to programming/erasure status (bit-rate selection complete) |
| H'31 | Erasing the user MAT or user boot MAT |
| H'3F | Waiting for programming/erasure selection (erasure complete) |
| H'4F | Waiting to receive data for programming (programming complete) |
| H'5F | Waiting for erasure block specification (erasure complete) |

**Table 22.16 Error Codes**

| Code | Description |
|---|---|
| H'00 | No error |
| H'11 | Sum check error |
| H'21 | Non-matching device code error |
| H'22 | Non-matching clock mode error |
| H'24 | Bit-rate selection failure |
| H'25 | Input frequency error |
| H'26 | Frequency multiplier error |
| H'27 | Operating frequency error |
| H'29 | Block number error |
| H'2A | Address error |
| H'2B | Data length error (size error) |
| H'51 | Erasure error |
| H'52 | Non-erased error |
| H'53 | Programming error |
| H'54 | Selection processing error |
| H'80 | Command error |
| H'FF | Bit-rate matching acknowledge error |

### 22.9.2 Areas for Storage of the Procedural Program and Data for Programming

In the descriptions in the previous section, storable areas for the programming/erasing procedure programs and program data are assumed to be in on-chip RAM. However, the procedure programs and data can be stored in and executed from other areas (e.g. external address space) as long as the following conditions are satisfied.

1. The on-chip programming/erasing program is downloaded from the address set by FTDAR in on-chip RAM, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use 128 bytes or more as a stack. Make sure this area is reserved.
3. Since download by setting the SCO bit to 1 will cause the MATs to be switched, it should be executed in on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been decided. When in a mode in which the external address space

is not accessible, such as single-chip mode, the required procedure programs, interrupt vector table, interrupt processing routine, and user branch program should be transferred to on-chip RAM before programming/erasing of the flash memory starts.

5. The flash memory is not accessible during programming/erasing operations. Therefore, the programming/erasing program must be downloaded to on-chip RAM in advance. Areas for executing each procedure program for initiating programming/erasing, the user program at the user branch destination for programming/erasing, the interrupt vector table, and the interrupt processing routine must be located in on-chip memory other than flash memory or the external address space.
6. After programming/erasing, access to flash memory is inhibited until FKEY is cleared.

   A reset state ($\overline{RES}$ = 0) for more than at least 100 μs must be taken when the LSI mode is changed to reset on completion of a programming/erasing operation.

   Transitions to the reset state during programming/erasing are inhibited. When the reset signal is accidentally input to the LSI, a longer period in the reset state than usual (100 μs) is needed before the reset signal is released.
7. Switching of the MATs by FMATS is needed for programming/erasing of the user MAT in user boot mode. The program which switches the MATs should be executed from the on-chip RAM. For details, see section 22.8.1, Switching between User MAT and User Boot MAT. Please make sure you know which MAT is selected when switching the MATs.
8. When the program data storage area indicated by the FMPDR parameter in the programming processing is within the flash memory area, an error will occur. Therefore, temporarily transfer the program data to on-chip RAM to change the address set in FMPDR to an address other than flash memory.

Based on these conditions, tables 22.17 and 22.18 show the areas in which the program data can be stored and executed according to the operation type and mode.

**Table 22.17 Executable MAT**

| | Initiated Mode | |
|---|---|---|
| **Operation** | **User Program Mode** | **User Boot Mode*** |
| Programming | Table 22.18 (1) | Table 22.18 (3) |
| Erasing | Table 22.18 (2) | Table 22.18 (4) |

Note: * Programming/Erasing is possible to user MATs.

**Table 22.18 (1) Usable Area for Programming in User Program Mode**

| | Item | Storable/Executable Area | | | Selected MAT | |
|---|---|---|---|---|---|---|
| | | On-Chip RAM | User MAT | External Space | User MAT | Embedded Program Storage MAT |
| Programming procedure | Program data storage area | √ | X* | √ | — | — |
| | Selecting on-chip program to be downloaded | √ | √ | √ | √ | |
| | Writing H'A5 to key register | √ | √ | √ | √ | |
| | Writing 1 to SCO in FCCS (download) | √ | X | X | | √ |
| | Key register clearing | √ | √ | √ | √ | |
| | Judging download result | √ | √ | √ | √ | |
| | Download error processing | √ | √ | √ | √ | |
| | Setting initialization parameters | √ | √ | √ | √ | |
| | Initialization | √ | X | X | √ | |
| | Judging initialization result | √ | √ | √ | √ | |
| | Initialization error processing | √ | √ | √ | √ | |
| | Interrupt processing routine | √ | X | √ | √ | |
| | Writing H'5A to key register | √ | √ | √ | √ | |
| | Setting programming parameters | √ | X | √ | √ | |
| | Programming | √ | X | X | √ | |
| | Judging programming result | √ | X | √ | √ | |
| | Programming error processing | √ | X | √ | √ | |
| | Key register clearing | √ | X | √ | √ | |

Note: * If the data has been transferred to on-chip RAM in advance, this area can be used.

**Table 22.18 (2) Usable Area for Erasure in User Program Mode**

| | | Storable/Executable Area | | | Selected MAT | |
|---|---|---|---|---|---|---|
| | **Item** | **On-Chip RAM** | **User MAT** | **External Space** | **User MAT** | **Embedded Program Storage MAT** |
| Erasing procedure | Selecting on-chip program to be downloaded | √ | √ | √ | √ | |
| | Writing H'A5 to key register | √ | √ | √ | √ | |
| | Writing 1 to SCO in FCCS (download) | √ | X | X | | √ |
| | Key register clearing | √ | √ | √ | √ | |
| | Judging download result | √ | √ | √ | √ | |
| | Download error processing | √ | √ | √ | √ | |
| | Setting initialization parameters | √ | √ | √ | √ | |
| | Initialization | √ | X | X | √ | |
| | Judging initialization result | √ | √ | √ | √ | |
| | Initialization error processing | √ | √ | √ | √ | |
| | Interrupt processing routine | √ | X | √ | √ | |
| | Writing H'5A to key register | √ | √ | √ | √ | |
| | Setting erasure parameters | √ | X | √ | √ | |
| | Erasure | √ | X | X | √ | |
| | Judging erasure result | √ | X | √ | √ | |
| | Erasing error processing | √ | X | √ | √ | |
| | Key register clearing | √ | X | √ | √ | |

**Table 22.18 (3) Usable Area for Programming in User Boot Mode**

| | | Storable/Executable Area | | | Selected MAT | | |
|---|---|---|---|---|---|---|---|
| | **Item** | **On-Chip RAM** | **User Boot MAT** | **External Space** | **User MAT** | **User Boot MAT** | **Embedded Program Storage Area** |
| Programming procedure | Program data storage area | √ | X*[1] | √ | — | — | — |
| | Selecting on-chip program to be downloaded | √ | √ | √ | | √ | |
| | Writing H'A5 to key register | √ | √ | √ | | √ | |
| | Writing 1 to SCO in FCCS (download) | √ | X | X | | | √ |
| | Key register clearing | √ | √ | √ | | √ | |
| | Judging download result | √ | √ | √ | | √ | |
| | Download error processing | √ | √ | √ | | √ | |
| | Setting initialization parameters | √ | √ | √ | | √ | |
| | Initialization | √ | X | X | | √ | |
| | Judging initialization result | √ | √ | √ | | √ | |
| | Initialization error processing | √ | √ | √ | | √ | |
| | Interrupt processing routine | √ | X | √ | | √ | |
| | Switching MATs by FMATS | √ | X | X | √ | | |
| | Writing H'5A to Key Register | √ | X | √ | √ | | |

**Table 22.18 (3) Usable Area for Programming in User Boot Mode (cont)**

| | | Storable/Executable Area | | | Selected MAT | | |
|---|---|---|---|---|---|---|---|
| | **Item** | **On-Chip RAM** | **User Boot MAT** | **External Space** | **User MAT** | **User Boot MAT** | **Embedded Program Storage Area** |
| Programming procedure | Setting programming parameters | √ | X | √ | √ | | |
| | Programming | √ | X | X | √ | | |
| | Judging programming result | √ | X | √ | √ | | |
| | Programming error processing | √ | X*2 | √ | √ | | |
| | Key register clearing | √ | X | √ | √ | | |
| | Switching MATs by FMATS | √ | X | X | | √ | |

Notes: 1. If the data has been transferred to on-chip RAM in advance, this area can be used.
2. If the MATs have been switched by FMATS in on-chip RAM, this MAT can be used.

**Table 22.18 (4) Usable Area for Erasure in User Boot Mode**

| | | Storable/Executable Area | | | Selected MAT | | |
|---|---|---|---|---|---|---|---|
| | **Item** | **On-Chip RAM** | **User Boot MAT** | **External Space** | **User MAT** | **User Boot MAT** | **Embedded Program Storage Area** |
| Erasing procedure | Selecting on-chip program to be downloaded | √ | √ | √ | | √ | |
| | Writing H'A5 to key register | √ | √ | √ | | √ | |
| | Writing 1 to SCO in FCCS (download) | √ | X | X | | | √ |
| | Key register clearing | √ | √ | √ | | √ | |
| | Judging download result | √ | √ | √ | | √ | |
| | Download error processing | √ | √ | √ | | √ | |
| | Setting initialization parameters | √ | √ | √ | | √ | |
| | Initialization | √ | X | X | | √ | |
| | Judging initialization result | √ | √ | √ | | √ | |
| | Initialization error processing | √ | √ | √ | | √ | |
| | Interrupt processing routine | √ | X | √ | | √ | |
| | Switching MATs by FMATS | √ | X | X | | √ | |
| | Writing H'5A to key register | √ | X | √ | √ | | |
| | Setting erasure parameters | √ | X | √ | √ | | |

**Table 22.18 (4) Usable Area for Erasure in User Boot Mode (cont)**

| | | Storable/Executable Area | | | Selected MAT | | |
|---|---|---|---|---|---|---|---|
| | **Item** | **On-Chip RAM** | **User Boot MAT** | **External Space** | **User MAT** | **User Boot MAT** | **Embedded Program Storage Area** |
| Erasing procedure | Erasure | √ | X | X | √ | | |
| | Judging erasure result | √ | X | √ | √ | | |
| | Erasing error processing | √ | X* | √ | √ | | |
| | Key register clearing | √ | X | √ | √ | | |
| | Switching MATs by FMATS | √ | X | X | | √ | |

Note: * If the MATs have been switched by FMATS in on-chip RAM, this MAT can be used.

## 22.10 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as for a discrete flash memory. Use a PROM programmer that supports the Renesas 128- or 256-Kbyte flash memory on-chip MCU device type (F-ZTATxxxx).

# Section 23 RAM

This LSI has an on-chip high-speed static RAM. The on-chip RAM is connected to the CPU by a 32-bit data bus (L bus), and to the data transfer controller (DTC) by a 32-bit data bus (I bus), enabling 8, 16, or 32-bit width access to data in the on-chip RAM.

The on-chip RAM is allocated to different addresses according to each product as shown in figure 23.1, and the on-chip RAM is divided into page 0 and page 1 based on the addresses. The on-chip RAM can be accessed from the CPU (via the L bus) and DTC (via the I bus). When different buses request to access the same page simultaneously, the priority becomes I bus (DTC) > L bus (CPU). Since such kind of conflict degrades the RAM access performance, software should be created so as to avoid conflicts. For example, conflict does not occur when the buses access different pages. An access from the L bus (CPU) is a 1-cycle access as long as page conflict does not occur. The number of bus cycles in accesses from the I bus (DTC) differ depending on the ratio between the internal clock (Iϕ) and bus clock (Bϕ), and the operating state of the DTC. The contents of the on-chip RAM are retained in sleep mode or software standby mode, and at a power-on reset or manual reset. However, the contents of the on-chip RAM are not retained in deep software standby mode.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the RAM control register (RAMCR). For details on the RAM control register (RAMCR), refer to section 24.3.7, RAM Control Register (RAMCR).

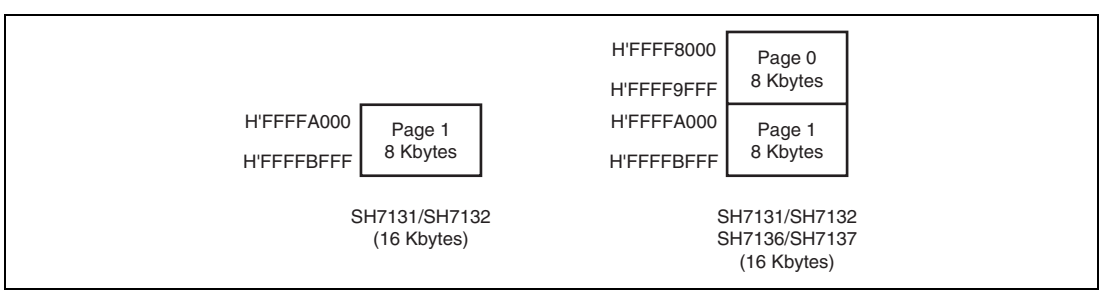



**Figure 23.1 On-chip RAM Addresses**

## 23.1 Usage Notes

### 23.1.1 Module Standby Mode Setting

RAM can be enabled/disabled by the standby control register. The initial value enables RAM operation. RAM access is disabled by setting the module standby mode. For details, see section 24, Power-Down Modes.

### 23.1.2 Address Error

When an address error in write access to the on-chip RAM occurs, the contents of the on-chip RAM may be corrupted.

### 23.1.3 Initial Values in RAM

After power has been supplied, initial values in RAM remain undefined until RAM is written.

# Section 24 Power-Down Modes

This LSI supports the following power-down modes: sleep mode, software standby mode, deep software standby mode, and module standby mode.

## 24.1 Features

- Supports sleep mode, software standby mode, module standby mode, and deep software standby mode.

### 24.1.1 Types of Power-Down Modes

This LSI has the following power-down modes.

- Sleep mode
- Software standby mode (SH7136 and SH7137 only)
- Deep software standby mode (SH7136 and SH7137 only)
- Module standby mode

Table 24.1 shows the methods to make a transition 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 24.1 States of Power-Down Modes**

| | | State | | | | | |
|---|---|---|---|---|---|---|---|
| **Mode** | **Transition Method** | **CPG** | **CPU** | **CPU Register** | **On-Chip Memory** | **On-Chip Peripheral Modules** | **Canceling Procedure** |
| Sleep | Execute SLEEP instruction with STBY bit in STBCR1 cleared to 0. | Runs | Halts | Held | Runs | Run | • Reset |
| Software standby* | Execute SLEEP instruction with STBY bit in STBCR1 and STBYMD bit in STBCR6 set to 1. | Halts | Halts | Held | Halts (contents retained) | Halt | • Interrupt by NMI or IRQ<br>• Power-on reset by the $\overline{\text{RES}}$ pin |
| Deep software standby* | Execute SLEEP instruction with STBY bit in STBCR1 set to 1 and STBYMD bit in STBCR6 cleared to 0. | Halts | Halts | Undefined | Halts (contents undefined) | Halt | • Power-on reset by the $\overline{\text{RES}}$ pin |
| Module standby | Set MSTP bits in STBCR2 to STBCR5 to 1. | Runs | Runs | Held | Specified module halts (contents retained) | Specified module halts | • Clear MSTP bit to 0<br>• Power-on reset (for modules whose MSTP bit has an initial value of 0) |

Notes: For details on the states of on-chip peripheral module registers in each mode, refer to section 25.3, Register States in Each Operating Mode. For details on the pin states in each mode, refer to appendix A, Pin States.

* SH7136 and SH7137 only.

## 24.2 Input/Output Pins

Table 24.2 lists the pins used for the power-down modes.

**Table 24.2 Pin Configuration**

| Pin Name | Symbol | I/O | Description |
|---|---|---|---|
| Power-on reset | $\overline{\text{RES}}$ | Input | Power-on reset input signal. Power-on reset by low level. |
| Manual reset | $\overline{\text{MRES}}$ | Input | Manual reset input signal. Manual reset by low level. |

## 24.3 Register Descriptions

There are following registers used for the power-down modes. For details on the addresses of these registers and the states of these registers in each processing state, see section 25, List of Registers.

**Table 24.3 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Standby control register 1 | STBCR1 | R/W | H'00 | H'FFFFE802 | 8 |
| Standby control register 2 | STBCR2 | R/W | H'38 | H'FFFFE804 | 8 |
| Standby control register 3 | STBCR3 | R/W | H'FF | H'FFFFE806 | 8 |
| Standby control register 4 | STBCR4 | R/W | H'FF | H'FFFFE808 | 8 |
| Standby control register 5 | STBCR5 | R/W | H'03 | H'FFFFE80A | 8 |
| Standby control register 6 | STBCR6 | R/W | H'00 | H'FFFFE80C | 8 |
| RAM control register | RAMCR | R/W | H'10 | H'FFFFE880 | 8 |

### 24.3.1 Standby Control Register 1 (STBCR1)

STBCR1 is an 8-bit readable/writable register that specifies the state of the power-down mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | STBY | - | - | - | - | - | - | - |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | STBY | 0 | R/W | Standby<br>Specifies transition to software standby mode.<br>0: Executing SLEEP instruction makes this LSI sleep mode<br>1: Executing SLEEP instruction makes this LSI software standby mode or deep software standby mode |
| 6 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

### 24.3.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MSTP 7 | MSTP 6 | - | MSTP 4 | - | - | - | - |
| Initial value: | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R | R/W | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | MSTP7 | 0 | R/W | Module Stop Bit 7<br>When this bit is set to 1, the clock supply to the RAM is halted.<br>0: RAM operates<br>1: Clock supply to RAM halted |
| 6 | MSTP6 | 0 | R/W | Module Stop Bit 6<br>When this bit is set to 1, the clock supply to the ROM is halted.<br>0: ROM operates<br>1: Clock supply to ROM halted |
| 5 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 4 | MSTP4 | 1 | R/W | Module Stop Bit 4<br>When this bit is set to 1, the clock supply to the DTC is halted.<br>0: DTC operates<br>1: Clock supply to the DTC halted |
| 3 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 2 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

### 24.3.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MSTP 15 | - | MSTP 13 | MSTP 12 | MSTP 11 | MSTP 10 | - | MSTP8 |
| Initial value: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W: | R/W | R | R/W | R/W | R/W | R/W | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | MSTP15 | 1 | R/W | Module Stop Bit 15<br>When this bit is set to 1, the clock supply to the $I^2C2$ is halted.<br>0: $I^2C2$ operates<br>1: Clock supply to $I^2C2$ halted |
| 6 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 5 | MSTP13 | 1 | R/W | Module Stop Bit 13<br>When this bit is set to 1, the clock supply to the SCI_2 is halted.<br>0: SCI_2 operates<br>1: Clock supply to SCI_2 halted |
| 4 | MSTP12 | 1 | R/W | Module Stop Bit 12<br>When this bit is set to 1, the clock supply to the SCI_1 is halted.<br>0: SCI_1 operates<br>1: Clock supply to SCI_1 halted |
| 3 | MSTP11 | 1 | R/W | Module Stop Bit 11<br>When this bit is set to 1, the clock supply to the SCI_0 is halted.<br>0: SCI_0 operates<br>1: Clock supply to SCI_0 halted |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | MSTP10 | 1 | R/W | Module Stop Bit 10<br>When this bit is set to 1, the clock supply to the SSU is halted.<br>0: SSU operates<br>1: Clock supply to SSU halted |
| 1 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 0 | MSTP8 | 1 | R/W | Module Stop Bit 8<br>When this bit is set to 1, the clock supply to the RCAN-ET_0 is halted.<br>0: RCAN-ET_0 operates<br>1: Clock supply to RCAN-ET_0 halted |

### 24.3.4 Standby Control Register 4 (STBCR4)

STBCR4 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MSTP 23 | MSTP 22 | MSTP 21 | MSTP 20 | MSTP 19 | - | - | - |
| 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 | MSTP23 | 1 | R/W | Module Stop Bit 23<br>When this bit is set to 1, the clock supply to the MTU2S is halted.<br>0: MTU2S operates<br>1: Clock supply to MTU2S halted |
| 6 | MSTP22 | 1 | R/W | Module Stop Bit 22<br>When this bit is set to 1, the clock supply to the MTU2 is halted.<br>0: MTU2 operates<br>1: Clock supply to MTU2 halted |
| 5 | MSTP21 | 1 | R/W | Module Stop Bit 21<br>When this bit is set to 1, the clock supply to the CMT is halted.<br>0: CMT operates<br>1: Clock supply to CMT halted |
| 4 | MSTP20 | 1 | R/W | Module Stop Bit 20<br>When this bit is set to 1, the clock supply to the A/D_1 is halted.<br>0: A/D_1 operates<br>1: Clock supply to A/D_1 halted |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | MSTP19 | 1 | R/W | Module Stop Bit 19<br>When this bit is set to 1, the clock supply to the A/D_0 is halted.<br>0: A/D_0 operates<br>1: Clock supply to A/D_0 halted |
| 2 to 0 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. |

### 24.3.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | MSTP 25 | MSTP 24 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W: | R | R | R | R | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | MSTP25 | 1 | R/W | Module Stop Bit 25<br>When this bit is set to 1, the clock supply to the AUD is halted.<br>0: AUD operates<br>1: Clock supply to AUD halted |
| 0 | MSTP24 | 1 | R/W | Module Stop Bit 24<br>When this bit is set to 1, the clock supply to the UBC is halted.<br>0: UBC operates<br>1: Clock supply to UBC halted |

### 24.3.6 Standby Control Register 6 (STBCR6)

STBCR6 is an 8-bit readable/writable register that specifies the state of the power-down modes.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | AUD SRST | HIZ | - | - | - | - | STBY MD | - |
| 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 | AUDSRST | 0 | R/W | AUD Software Reset<br>This bit controls the AUD reset by software. When 0 is written to AUDSRST, the AUD module shifts to the power-on reset state.<br>0: Shifts to the AUD reset state<br>1: Clears the AUD reset<br>When setting this bit to 1, MSTP25 in STBCR5 should be 0. |
| 6 | HIZ | 0 | R/W | Port High-Impedance<br>In software standby mode, this bit selects whether the pin state is retained or changed to high-impedance.<br>0: In software standby mode, the pin state is retained<br>1: In software standby mode, the pin state is changed to high-impedance |
| 5 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | STBYMD | 0 | R/W | Software Standby Mode Select<br>This bit selects a transition to software standby mode or deep software standby mode by executing the SLEEP instruction when the STBY bit is 1 in STBCR1.<br>0: Transition to deep software standby mode<br>1: Transition to software standby mode |
| 0 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

### 24.3.7 RAM Control Register (RAMCR)

RAMCR is an 8-bit readable/writable register that enables/disables the access to the on-chip RAM.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | RAME | - | - | - | - |
| Initial value: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R/W | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 4 | RAME | 1 | R/W | RAM Enable<br>This bit enables/disables the on-chip RAM.<br>0: On-chip RAM disabled<br>1: On-chip RAM enabled<br>When this bit is cleared to 0, the access to the on-chip RAM is disabled. In this case, an undefined value is returned when reading or fetching the data or instruction from the on-chip RAM, and writing to the on-chip RAM is ignored.<br>When RAME is cleared to 0 to disable the on-chip RAM, an instruction to access the on-chip RAM should not be set next to the instruction to write RAMCR. If such an instruction is set, normal access is not guaranteed.<br>When RAME is set to 1 to enable the on-chip RAM, an instruction to read RAMCR should be set next to the instruction to write to RAMCR. If an instruction to access the on-chip RAM is set next to the instruction to write to RAMCR, normal access is not guaranteed. |
| 3 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

## 24.4 Sleep Mode

### 24.4.1 Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR1 is 0 causes a transition from the program execution state to sleep mode. However, sleep mode cannot be entered when the bus is released (low-level input to $\overline{\text{BREQ}}$ pin). Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to operate.

### 24.4.2 Canceling Sleep Mode

Sleep mode is canceled by a reset.

Do not cancel sleep mode with an interrupt.

**Canceling with Reset:** Sleep mode is canceled by a power-on reset with the $\overline{\text{RES}}$ pin, a manual reset with the $\overline{\text{MRES}}$ pin, or an internal power-on/manual reset by WDT.

## 24.5 Software Standby Mode (SH7136 and SH7137 only)

### 24.5.1 Transition to Software Standby Mode

This LSI switches from a program execution state to software standby mode by executing the SLEEP instruction when the STBY bit in STBCR1 and the STBYMD bit in STBCR6 are set to 1. However, software standby mode cannot be entered when the bus is released (low-level input to $\overline{\text{BREQ}}$ pin). Execute the SLEEP instruction after halting the DTC. In software standby mode, not only the CPU but also the clock and on-chip peripheral modules halt.

The contents of the CPU registers and the data of the on-chip RAM remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. For details on the states of on-chip peripheral module registers in software standby mode, refer to section 25.3, Register States in Each Operating Mode. For details on the pin states in software standby mode, refer to appendix A, Pin States.

The procedure for switching to software standby mode is as follows:

1. Clear the TME bit in the timer control register (WTCSR) of the WDT to 0 to stop the WDT.
2. Set the timer counter (WTCNT) of the WDT to 0 and bits CKS2 to CKS0 in WTCSR to appropriate values to secure the specified oscillation settling time.
3. If the DTC is operating, stop its operation.
4. If the bus is released (low-level input to $\overline{\text{BREQ}}$ pin), acquire the bus mastership (high-level input to $\overline{\text{BREQ}}$ pin).
5. After setting the STBY bit in STBCR1 and the STBYMD bit in STBCR6 to 1, execute the SLEEP instruction.
6. Software standby mode is entered and the clocks within this LSI are halted.

### 24.5.2 Canceling Software Standby Mode

Software standby mode is canceled by interrupts (NMI, IRQ) or a reset.

**Canceling with Interrupt:** The WDT can be used for hot starts. When an NMI or IRQ interrupt (edge detection) is detected, the clock will be supplied to the entire LSI and software standby mode will be canceled after the time set in the timer control/status register of the WDT has elapsed. Interrupt exception handling is then executed.

When the priority level of an IRQ interrupt is lower than the interrupt mask level set in the status register (SR) of the CPU, an interrupt request is not accepted preventing software standby mode from being canceled.

When falling-edge detection is selected for the NMI pin, drive the NMI pin high before making a transition to software standby mode. When rising-edge detection is selected for the NMI pin, drive the NMI pin low before making a transition to software standby mode.

Similarly, when falling-edge detection is selected for the IRQ pin, drive the IRQ pin high before making a transition to software standby mode. When rising-edge detection is selected for the IRQ pin, drive the IRQ pin low before making a transition to software standby mode.

**Canceling with Power-on Reset:** Software standby mode is canceled by a power-on reset with the $\overline{\text{RES}}$ pin. Keep the $\overline{\text{RES}}$ pin low until the clock oscillation settles.

## 24.6 Deep Software Standby Mode (SH7136 and SH7137 only)

### 24.6.1 Transition to Deep Software Standby Mode

This LSI shifts from a program execution state to deep software standby mode by executing the SLEEP instruction when the STBY bit in STBCR1 is 1 and the STBYMD bit in STBCR6 is 0. However, deep software standby mode cannot be entered when the bus is released (low-level input to $\overline{\text{BREQ}}$ pin). Execute the SLEEP instruction after halting the DTC. In deep software standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. Furthermore, the internal power supply of this LSI is turned off.

The contents of the CPU registers and the data of the on-chip RAM become undefined. The registers of on-chip peripheral modules are initialized. For details on the pin states in deep software standby mode, refer to appendix A, Pin States.

The procedure for a transition to deep software standby mode is as follows:

1. Clear the TME bit in the timer control register (WTCSR) of the WDT to 0 to stop the WDT.
2. If the DTC is operating, stop its operation.
3. If the bus is released (low-level input to $\overline{\text{BREQ}}$ pin), acquire the bus mastership (high-level input to $\overline{\text{BREQ}}$ pin).
4. After setting the STBY bit in STBCR1 to 1 and clearing the STBYMD bit in STBCR6 to 0, execute the SLEEP instruction.
5. Deep software standby mode is entered, the clocks within this LSI are halted, and the internal power supply of this LSI is turned off.

### 24.6.2 Canceling Deep Software Standby Mode

Deep software standby mode is canceled by a power-on reset with the $\overline{\text{RES}}$ pin. Keep the $\overline{\text{RES}}$ pin low until the clock oscillation settles.

## 24.7 Module Standby Mode

### 24.7.1 Transition to Module Standby Mode

Setting the MSTP bits in the standby control registers (STBCR2 to STBCR5) 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.

Do not access registers of an on-chip peripheral module which has been set to enter module standby mode. For details on the states of on-chip peripheral module registers in module standby mode, refer to section 25.3, Register States in Each Operating Mode.

### 24.7.2 Canceling Module Standby Function

The module standby function can be canceled by clearing the MSTP bits in STBCR2 to STBCR5 to 0. The module standby function can be canceled by a power-on reset for modules whose MSTP bit has an initial value of 0.

## 24.8 Usage Note

### 24.8.1 Current Consumption while Waiting for Oscillation to be Stabilized

The current consumption while waiting for oscillation to be stabilized is higher than that while oscillation is stabilized.

### 24.8.2 Executing the SLEEP Instruction

Apply either of the following measures before executing the SLEEP instruction to initiate the transition to sleep mode or software standby mode.

Measure A: Stop the operation of the DTC and the generation of interrupts from on-chip peripheral modules, IRQ interrupts, and the NMI interrupt before executing the SLEEP instruction.

Measure B: Change the value in FRQCR to the initial value, H'36DB, and then dummy-read FRQCR twice before executing the SLEEP instruction.

# Section 25 List of Registers

This section gives information on internal I/O registers. The contents of this section are as follows:

1. Register Address Table (in the order from a lower address)
   - Registers are listed in the order from lower allocated addresses.
   - As for reserved addresses, the register name column is indicated with —. Do not access reserved addresses.
   - As for 16- or 32-bit address, the MSB addresses are shown.
   - The list is classified according to module names.
   - The numbers of access cycles are given.

2. Register Bit Table
   - Bit configurations are shown in the order of the register address table.
   - As for reserved bits, the bit name column is indicated with —.
   - As for the blank column of the bit names, the whole register is allocated to the counter or data.
   - As for 16- or 32-bit registers, bits are indicated from the MSB.

3. Register State in Each Operating Mode
   - Register states are listed in the order of the register address table.
   - Register states in the basic operating mode are shown. As for modules including their specific states such as reset, see the sections of those modules.

## 25.1 Register Address Table (In the Order of Addresses)

Access sizes are indicated as the number of bits. Access cycles are the number of cycles of the indicated reference clock, and the values are shown for 8-bit access (B), 16-bit access (W), or 32-bit access (L).

Notes: 1. Access to undefined locations or reserved addresses is prohibited. Correct operation cannot be guaranteed if such addresses are accessed.
2. Access to mailbox areas of the RCAN-ET may include wait cycles of 0 to 5 Pϕ cycles.

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Serial mode register_0 | SCSMR_0 | 8 | H'FFFFC000 | SCI (Channel 0) | 8 | Pϕ (reference clock) B: 2 | 16 bits |
| Bit rate register_0 | SCBRR_0 | 8 | H'FFFFC002 | | 8 | | |
| Serial control register_0 | SCSCR_0 | 8 | H'FFFFC004 | | 8 | | |
| Transmit data register_0 | SCTDR_0 | 8 | H'FFFFC006 | | 8 | | |
| Serial status register_0 | SCSSR_0 | 8 | H'FFFFC008 | | 8 | | |
| Receive data register_0 | SCRDR_0 | 8 | H'FFFFC00A | | 8 | | |
| Serial direction control register_0 | SCSDCR_0 | 8 | H'FFFFC00C | | 8 | | |
| Serial port register_0 | SCSPTR_0 | 8 | H'FFFFC00E | | 8 | | |
| Serial mode register_1 | SCSMR_1 | 8 | H'FFFFC080 | SCI (Channel 1) | 8 | Pϕ (reference clock) B: 2 | 16 bits |
| Bit rate register_1 | SCBRR_1 | 8 | H'FFFFC082 | | 8 | | |
| Serial control register_1 | SCSCR_1 | 8 | H'FFFFC084 | | 8 | | |
| Transmit data register_1 | SCTDR_1 | 8 | H'FFFFC086 | | 8 | | |
| Serial status register_1 | SCSSR_1 | 8 | H'FFFFC088 | | 8 | | |
| Receive data register_1 | SCRDR_1 | 8 | H'FFFFC08A | | 8 | | |
| Serial direction control register_1 | SCSDCR_1 | 8 | H'FFFFC08C | | 8 | | |
| Serial port register_1 | SCSPTR_1 | 8 | H'FFFFC08E | | 8 | | |
| Serial mode register_2 | SCSMR_2 | 8 | H'FFFFC100 | SCI (Channel 2) | 8 | Pϕ (reference clock) B: 2 | 16 bits |
| Bit rate register_2 | SCBRR_2 | 8 | H'FFFFC102 | | 8 | | |
| Serial control register_2 | SCSCR_2 | 8 | H'FFFFC104 | | 8 | | |
| Transmit data register_2 | SCTDR_2 | 8 | H'FFFFC106 | | 8 | | |
| Serial status register_2 | SCSSR_2 | 8 | H'FFFFC108 | | 8 | | |
| Receive data register_2 | SCRDR_2 | 8 | H'FFFFC10A | | 8 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Serial direction control register_2 | SCSDCR_2 | 8 | H'FFFFC10C | SCI (Channel 2) | 8 | Pϕ (reference clock) B: 2 | 16 bits |
| Serial port register_2 | SCSPTR_2 | 8 | H'FFFFC10E | | 8 | | |
| Timer control register_3 | TCR_3 | 8 | H'FFFFC200 | MTU2 | 8, 16, 32 | MPϕ (reference clock) B: 2 W: 2 L: 4 | 16 bits |
| Timer control register_4 | TCR_4 | 8 | H'FFFFC201 | | 8 | | |
| Timer mode register_3 | TMDR_3 | 8 | H'FFFFC202 | | 8, 16 | | |
| Timer mode register_4 | TMDR_4 | 8 | H'FFFFC203 | | 8 | | |
| Timer I/O control register H_3 | TIORH_3 | 8 | H'FFFFC204 | | 8, 16, 32 | | |
| Timer I/O control register L_3 | TIORL_3 | 8 | H'FFFFC205 | | 8 | | |
| Timer I/O control register H_4 | TIORH_4 | 8 | H'FFFFC206 | | 8, 16 | | |
| Timer I/O control register L_4 | TIORL_4 | 8 | H'FFFFC207 | | 8 | | |
| Timer interrupt enable register_3 | TIER_3 | 8 | H'FFFFC208 | | 8, 16 | | |
| Timer interrupt enable register_4 | TIER_4 | 8 | H'FFFFC209 | | 8 | | |
| Timer output master enable register | TOER | 8 | H'FFFFC20A | | 8 | | |
| Timer gate control register | TGCR | 8 | H'FFFFC20D | | 8 | | |
| Timer output control register 1 | TOCR1 | 8 | H'FFFFC20E | | 8, 16 | | |
| Timer output control register 2 | TOCR2 | 8 | H'FFFFC20F | | 8 | | |
| Timer counter_3 | TCNT_3 | 16 | H'FFFFC210 | | 16, 32 | | |
| Timer counter_4 | TCNT_4 | 16 | H'FFFFC212 | | 16 | | |
| Timer cycle data register | TCDR | 16 | H'FFFFC214 | | 16, 32 | | |
| Timer dead time data register | TDDR | 16 | H'FFFFC216 | | 16 | | |
| Timer general register A_3 | TGRA_3 | 16 | H'FFFFC218 | | 16, 32 | | |
| Timer general register B_3 | TGRB_3 | 16 | H'FFFFC21A | | 16 | | |
| Timer general register A_4 | TGRA_4 | 16 | H'FFFFC21C | | 16, 32 | | |
| Timer general register B_4 | TGRB_4 | 16 | H'FFFFC21E | | 16 | | |
| Timer sub-counter | TCNTS | 16 | H'FFFFC220 | | 16, 32 | | |
| Timer cycle buffer register | TCBR | 16 | H'FFFFC222 | | 16 | | |
| Timer general register C_3 | TGRC_3 | 16 | H'FFFFC224 | | 16, 32 | | |
| Timer general register D_3 | TGRD_3 | 16 | H'FFFFC226 | | 16 | | |
| Timer general register C_4 | TGRC_4 | 16 | H'FFFFC228 | | 16, 32 | | |
| Timer general register D_4 | TGRD_4 | 16 | H'FFFFC22A | | 16 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Timer status register_3 | TSR_3 | 8 | H'FFFFC22C | MTU2 | 8, 16 | MPϕ (reference clock)<br>B: 2<br>W: 2<br>L: 4 | 16 bits |
| Timer status register_4 | TSR_4 | 8 | H'FFFFC22D | | 8 | | |
| Timer interrupt skipping set register | TITCR | 8 | H'FFFFC230 | | 8, 16 | | |
| Timer interrupt skipping counter | TITCNT | 8 | H'FFFFC231 | | 8 | | |
| Timer buffer transfer set register | TBTER | 8 | H'FFFFC232 | | 8 | | |
| Timer dead time enable register | TDER | 8 | H'FFFFC234 | | 8 | | |
| Timer output level buffer register | TOLBR | 8 | H'FFFFC236 | | 8 | | |
| Timer buffer operation transfer mode register_3 | TBTM_3 | 8 | H'FFFFC238 | | 8, 16 | | |
| Timer buffer operation transfer mode register_4 | TBTM_4 | 8 | H'FFFFC239 | | 8 | | |
| Timer A/D converter start request control register | TADCR | 16 | H'FFFFC240 | | 16 | | |
| Timer A/D converter start request cycle set register A_4 | TADCORA_4 | 16 | H'FFFFC244 | | 16, 32 | | |
| Timer A/D converter start request cycle set register B_4 | TADCORB_4 | 16 | H'FFFFC246 | | 16 | | |
| Timer A/D converter start request cycle set buffer register A_4 | TADCOBRA_4 | 16 | H'FFFFC248 | | 16, 32 | | |
| Timer A/D converter start request cycle set buffer register B_4 | TADCOBRB_4 | 16 | H'FFFFC24A | | 16 | | |
| Timer waveform control register | TWCR | 8 | H'FFFFC260 | | 8 | | |
| Timer start register | TSTR | 8 | H'FFFFC280 | | 8, 16 | | |
| Timer synchronous register | TSYR | 8 | H'FFFFC281 | | 8 | | |
| Timer counter synchronous start register | TCSYSTR | 8 | H'FFFFC282 | | 8 | | |
| Timer read/write enable register | TRWER | 8 | H'FFFFC284 | | 8 | | |
| Timer control register_0 | TCR_0 | 8 | H'FFFFC300 | | 8, 16, 32 | | |
| Timer mode register_0 | TMDR_0 | 8 | H'FFFFC301 | | 8 | | |
| Timer I/O control register H_0 | TIORH_0 | 8 | H'FFFFC302 | | 8, 16 | | |
| Timer I/O control register L_0 | TIORL_0 | 8 | H'FFFFC303 | | 8 | | |
| Timer interrupt enable register_0 | TIER_0 | 8 | H'FFFFC304 | | 8, 16, 32 | | |
| Timer status register_0 | TSR_0 | 8 | H'FFFFC305 | | 8 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Timer counter_0 | TCNT_0 | 16 | H'FFFFC306 | MTU2 | 16 | MPϕ (reference clock) | 16 bits |
| Timer general register A_0 | TGRA_0 | 16 | H'FFFFC308 | | 16, 32 | B: 2 | |
| Timer general register B_0 | TGRB_0 | 16 | H'FFFFC30A | | 16 | W: 2 | |
| Timer general register C_0 | TGRC_0 | 16 | H'FFFFC30C | | 16, 32 | L: 4 | |
| Timer general register D_0 | TGRD_0 | 16 | H'FFFFC30E | | 16 | | |
| Timer general register E_0 | TGRE_0 | 16 | H'FFFFC320 | | 16, 32 | | |
| Timer general register F_0 | TGRF_0 | 16 | H'FFFFC322 | | 16 | | |
| Timer interrupt enable register 2_0 | TIER2_0 | 8 | H'FFFFC324 | | 8, 16 | | |
| Timer status register 2_0 | TSR2_0 | 8 | H'FFFFC325 | | 8 | | |
| Timer buffer operation transfer mode register_0 | TBTM_0 | 8 | H'FFFFC326 | | 8 | | |
| Timer control register_1 | TCR_1 | 8 | H'FFFFC380 | | 8, 16 | | |
| Timer mode register_1 | TMDR_1 | 8 | H'FFFFC381 | | 8 | | |
| Timer I/O control register_1 | TIOR_1 | 8 | H'FFFFC382 | | 8 | | |
| Timer interrupt enable register_1 | TIER_1 | 8 | H'FFFFC384 | | 8, 16, 32 | | |
| Timer status register_1 | TSR_1 | 8 | H'FFFFC385 | | 8 | | |
| Timer counter_1 | TCNT_1 | 16 | H'FFFFC386 | | 16 | | |
| Timer general register A_1 | TGRA_1 | 16 | H'FFFFC388 | | 16, 32 | | |
| Timer general register B_1 | TGRB_1 | 16 | H'FFFFC38A | | 16 | | |
| Timer input capture control register | TICCR | 8 | H'FFFFC390 | | 8 | | |
| Timer control register_2 | TCR_2 | 8 | H'FFFFC400 | | 8, 16 | | |
| Timer mode register_2 | TMDR_2 | 8 | H'FFFFC401 | | 8 | | |
| Timer I/O control register_2 | TIOR_2 | 8 | H'FFFFC402 | | 8 | | |
| Timer interrupt enable register_2 | TIER_2 | 8 | H'FFFFC404 | | 8, 16, 32 | | |
| Timer status register_2 | TSR_2 | 8 | H'FFFFC405 | | 8 | | |
| Timer counter_2 | TCNT_2 | 16 | H'FFFFC406 | | 16 | | |
| Timer general register A_2 | TGRA_2 | 16 | H'FFFFC408 | | 16, 32 | | |
| Timer general register B_2 | TGRB_2 | 16 | H'FFFFC40A | | 16 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Timer counter U_5 | TCNTU_5 | 16 | H'FFFFC480 | MTU2 | 16, 32 | MPϕ (reference clock) | 16 bits |
| Timer general register U_5 | TGRU_5 | 16 | H'FFFFC482 | | 16 | B: 2 | |
| Timer control register U_5 | TCU_5 | 8 | H'FFFFC484 | | 8 | W: 2 | |
| | | | | | | L: 4 | |
| Timer I/O control register U_5 | TIORU_5 | 8 | H'FFFFC486 | | 8 | | |
| Timer counter V_5 | TCNTV_5 | 16 | H'FFFFC490 | | 16, 32 | | |
| Timer general register V_5 | TGRV_5 | 16 | H'FFFFC492 | | 16 | | |
| Timer control register V_5 | TCRV_5 | 8 | H'FFFFC494 | | 8 | | |
| Timer I/O control register V_5 | TIORV_5 | 8 | H'FFFFC496 | | 8 | | |
| Timer counter W_5 | TCNTW_5 | 16 | H'FFFFC4A0 | | 16, 32 | | |
| Timer general register W_5 | TGRW_5 | 16 | H'FFFFC4A2 | | 16 | | |
| Timer control register W_5 | TCRW_5 | 8 | H'FFFFC4A4 | | 8 | | |
| Timer I/O control register W_5 | TIORW_5 | 8 | H'FFFFC4A6 | | 8 | | |
| Timer status register_5 | TSR_5 | 8 | H'FFFFC4B0 | | 8 | | |
| Timer interrupt enable register_5 | TIER_5 | 8 | H'FFFFC4B2 | | 8 | | |
| Timer start register_5 | TSTR_5 | 8 | H'FFFFC4B4 | | 8 | | |
| Timer compare match clear register | TCNTCMPCLR | 8 | H'FFFFC4B6 | | 8 | | |
| Timer control register_3S | TCR_3S | 8 | H'FFFFC600 | | 8, 16, 32 | | |
| Timer control register_4S | TCR_4S | 8 | H'FFFFC601 | | 8 | | |
| Timer mode register_3S | TMDR_3S | 8 | H'FFFFC602 | | 8, 16 | | |
| Timer mode register_4S | TMDR_4S | 8 | H'FFFFC603 | | 8 | | |
| Timer I/O control register H_3S | TIORH_3S | 8 | H'FFFFC604 | | 8, 16, 32 | | |
| Timer I/O control register L_3S | TIORL_3S | 8 | H'FFFFC605 | | 8 | | |
| Timer I/O control register H_4S | TIORH_4S | 8 | H'FFFFC606 | | 8, 16 | | |
| Timer I/O control register L_4S | TIORL_4S | 8 | H'FFFFC607 | | 8 | | |
| Timer interrupt enable register_3S | TIER_3S | 8 | H'FFFFC608 | | 8, 16 | | |
| Timer interrupt enable register_4S | TIER_4S | 8 | H'FFFFC609 | | 8 | | |
| Timer output master enable register S | TOERS | 8 | H'FFFFC60A | | 8 | | |
| Timer gate control register S | TGCRS | 8 | H'FFFFC60D | | 8 | | |
| Timer output control register 1S | TOCR1S | 8 | H'FFFFC60E | | 8, 16 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Timer output control register 2S | TOCR2S | 8 | H'FFFFC60F | MTU2S | 8 | MIϕ (reference clock) | 16 bits |
| Timer counter_3S | TCNT_3S | 16 | H'FFFFC610 | | 16, 32 | B: 2 | |
| Timer counter_4S | TCNT_4S | 16 | H'FFFFC612 | | 16 | W: 2 | |
| Timer cycle data register S | TCDRS | 16 | H'FFFFC614 | | 16, 32 | L: 4 | |
| Timer dead time data register S | TDDRS | 16 | H'FFFFC616 | | 16 | | |
| Timer general register A_3S | TGRA_3S | 16 | H'FFFFC618 | | 16, 32 | | |
| Timer general register B_3S | TGRB_3S | 16 | H'FFFFC61A | | 16 | | |
| Timer general register A_4S | TGRA_4S | 16 | H'FFFFC61C | | 16, 32 | | |
| Timer general register B_4S | TGRB_4S | 16 | H'FFFFC61E | | 16 | | |
| Timer sub-counter S | TCNTSS | 16 | H'FFFFC620 | | 16, 32 | | |
| Timer cycle buffer register S | TCBRS | 16 | H'FFFFC622 | | 16 | | |
| Timer general register C_3S | TGRC_3S | 16 | H'FFFFC624 | | 16, 32 | | |
| Timer general register D_3S | TGRD_3S | 16 | H'FFFFC626 | | 16 | | |
| Timer general register C_4S | TGRC_4S | 16 | H'FFFFC628 | | 16, 32 | | |
| Timer general register D_4S | TGRD_4S | 16 | H'FFFFC62A | | 16 | | |
| Timer status register_3S | TSR_3S | 8 | H'FFFFC62C | | 8, 16 | | |
| Timer status register_4S | TSR_4S | 8 | H'FFFFC62D | | 8 | | |
| Timer interrupt skipping set register S | TITCRS | 8 | H'FFFFC630 | | 8, 16 | | |
| Timer interrupt skipping counter S | TITCNTS | 8 | H'FFFFC631 | | 8 | | |
| Timer buffer transfer set register S | TBTERS | 8 | H'FFFFC632 | | 8 | | |
| Timer dead time enable register S | TDERS | 8 | H'FFFFC634 | | 8 | | |
| Timer output level buffer register S | TOLBRS | 8 | H'FFFFC636 | | 8 | | |
| Timer buffer operation transfer mode register_3S | TBTM_3S | 8 | H'FFFFC638 | | 8, 16 | | |
| Timer buffer operation transfer mode register_4S | TBTM_4S | 8 | H'FFFFC639 | | 8 | | |
| Timer A/D converter start request control register S | TADCRS | 16 | H'FFFFC640 | | 16 | | |
| Timer A/D converter start request cycle set register A_4S | TADCORA_4S | 16 | H'FFFFC644 | | 16, 32 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Timer A/D converter start request cycle set register B_4S | TADCORB_4S | 16 | H'FFFFC646 | MTU2S | 16 | Mlϕ (reference clock) B: 2 W: 2 L: 4 | 16 bits |
| Timer A/D converter start request cycle set buffer register A_4S | TADCOBRA_4S | 16 | H'FFFFC648 | | 16, 32 | | |
| Timer A/D converter start request cycle set buffer register B_4S | TADCOBRB_4S | 16 | H'FFFFC64A | | 16 | | |
| Timer synchronous clear register S | TSYCRS | 8 | H'FFFFC650 | | 8 | | |
| Timer waveform control register S | TWCRS | 8 | H'FFFFC660 | | 8 | | |
| Timer start register S | TSTRS | 8 | H'FFFFC680 | | 8, 16 | | |
| Timer synchronous register S | TSYRS | 8 | H'FFFFC681 | | 8 | | |
| Timer read/write enable register S | TRWERS | 8 | H'FFFFC684 | | 8 | | |
| Timer counter U_5S | TCNTU_5S | 16 | H'FFFFC880 | | 16, 32 | | |
| Timer general register U_5S | TGRU_5S | 16 | H'FFFFC882 | | 16 | | |
| Timer control register U_5S | TCRU_5S | 8 | H'FFFFC884 | | 8 | | |
| Timer I/O control register U_5S | TIORU_5S | 8 | H'FFFFC886 | | 8 | | |
| Timer counter V_5S | TCNTV_5S | 16 | H'FFFFC890 | | 16, 32 | | |
| Timer general register V_5S | TGRV_5S | 16 | H'FFFFC892 | | 16 | | |
| Timer control register V_5S | TCRV_5S | 8 | H'FFFFC894 | | 8 | | |
| Timer I/O control register V_5S | TIORV_5S | 8 | H'FFFFC896 | | 8 | | |
| Timer counter W_5S | TCNTW_5S | 16 | H'FFFFC8A0 | | 16, 32 | | |
| Timer general register W_5S | TGRW_5S | 16 | H'FFFFC8A2 | | 16 | | |
| Timer control register W_5S | TCRW_5S | 8 | H'FFFFC8A4 | | 8 | | |
| Timer I/O control register W_5S | TIORW_5S | 8 | H'FFFFC8A6 | | 8 | | |
| Timer status register_5S | TSR_5S | 8 | H'FFFFC8B0 | | 8 | | |
| Timer interrupt enable register_5S | TIER_5S | 8 | H'FFFFC8B2 | | 8 | | |
| Timer start register_5S | TSTR_5S | 8 | H'FFFFC8B4 | | 8 | | |
| Timer compare match clear register S | TCNTCMPCLRS | 8 | H'FFFFC8B6 | | 8 | | |
| Flash code control/status register | FCCS | 8 | H'FFFFCC00 | FLASH | 8 | Pϕ (reference clock) B: 5 | 16 bits |
| Flash program code select register | FPCS | 8 | H'FFFFCC01 | | 8 | | |
| Flash erase code select register | FECS | 8 | H'FFFFCC02 | | 8 | | |
| Flash key code register | FKEY | 8 | H'FFFFCC04 | | 8 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Flash MAT select register | FMATS | 8 | H'FFFFCC05 | FLASH | 8 | Pϕ (reference clock) | 16 bits |
| Flash transfer destination address register | FTDAR | 8 | H'FFFFCC06 | | 8 | B: 5 | |
| DTC enable register A | DTCERA | 16 | H'FFFFCC80 | DTC | 8, 16 | Pϕ (reference clock) | 16 bits |
| DTC enable register B | DTCERB | 16 | H'FFFFCC82 | | 8, 16 | B: 2 | |
| DTC enable register C | DTCERC | 16 | H'FFFFCC84 | | 8, 16 | W: 2 | |
| DTC enable register D | DTCERD | 16 | H'FFFFCC86 | | 8, 16 | L: 4 | |
| DTC enable register E | DTCERE | 16 | H'FFFFCC88 | | 8, 16 | | |
| DTC control register | DTCCR | 8 | H'FFFFCC90 | | 8 | | |
| DTC vector base register | DTCVBR | 32 | H'FFFFCC94 | | 8, 16, 32 | | |
| $I^2C$ bus control register 1 | ICCR1 | 8 | H'FFFFCD80 | $I^2C2$ | 8 | Pϕ reference | 8 bits |
| $I^2C$ bus control register 2 | ICCR2 | 8 | H'FFFFCD81 | | 8 | B: 2 | |
| $I^2C$ bus mode register | ICMR | 8 | H'FFFFCD82 | | 8 | | |
| $I^2C$ bus interrupt enable register | ICIER | 8 | H'FFFFCD83 | | 8 | | |
| $I^2C$ bus status register | ICSR | 8 | H'FFFFCD84 | | 8 | | |
| Slave address register | SAR | 8 | H'FFFFCD85 | | 8 | | |
| $I^2C$ bus transmit data register | ICDRT | 8 | H'FFFFCD86 | | 8 | | |
| $I^2C$ bus receive data register | ICDRR | 8 | H'FFFFCD87 | | 8 | | |
| NF2CYC register | NF2CYC | 8 | H'FFFFCD88 | | 8 | | |
| SS control register H | SSCRH | 8 | H'FFFFCD00 | SSU | 8, 16 | Pϕ (reference clock) | 16 bits |
| SS control register L | SSCRL | 8 | H'FFFFCD01 | | 8 | B: 2 | |
| SS mode register | SSMR | 8 | H'FFFFCD02 | | 8, 16 | W: 2 | |
| SS enable register | SSER | 8 | H'FFFFCD03 | | 8 | | |
| SS status register | SSSR | 8 | H'FFFFCD04 | | 8, 16 | | |
| SS control register 2 | SSCR2 | 8 | H'FFFFCD05 | | 8 | | |
| SS transmit data register 0 | SSTDR0 | 8 | H'FFFFCD06 | | 8, 16 | | |
| SS transmit data register 1 | SSTDR1 | 8 | H'FFFFCD07 | | 8 | | |
| SS transmit data register 2 | SSTDR2 | 8 | H'FFFFCD08 | | 8, 16 | | |
| SS transmit data register 3 | SSTDR3 | 8 | H'FFFFCD09 | SSU | 8 | Pϕ (reference clock) | 16 bits |
| SS receive data register 0 | SSRDR0 | 8 | H'FFFFCD0A | | 8, 16 | B: 2 | |
| SS receive data register 1 | SSRDR1 | 8 | H'FFFFCD0B | | 8 | W: 2 | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| SS receive data register 2 | SSRDR2 | 8 | H'FFFFCD0C | SSU | 8, 16 | Pϕ (reference clock) | 16 bits |
| SS receive data register 3 | SSRDR3 | 8 | H'FFFFCD0D | | 8 | B: 2<br>W: 2 | |
| Compare match timer start register | CMSTR | 16 | H'FFFFCE00 | CMT | 8, 16, 32 | Pϕ (reference clock) | 16 bits |
| Compare match timer control/status register_0 | CMCSR_0 | 16 | H'FFFFCE02 | | 8, 16 | B: 2<br>W: 2 | |
| Compare match counter_0 | CMCNT_0 | 16 | H'FFFFCE04 | | 8, 16, 32 | L: 4 | |
| Compare match constant register_0 | CMCOR_0 | 16 | H'FFFFCE06 | | 8, 16 | | |
| Compare match timer control/status register_1 | CMCSR_1 | 16 | H'FFFFCE08 | | 8, 16, 32 | | |
| Compare match counter_1 | CMCNT_1 | 16 | H'FFFFCE0A | | 8, 16 | | |
| Compare match constant register_1 | CMCOR_1 | 16 | H'FFFFCE0C | | 8, 16, 32 | | |
| Input level control/status register 1 | ICSR1 | 16 | H'FFFFD000 | POE | 8, 16, 32 | Pϕ (reference clock) | 16 bits |
| Output level control/status register 1 | OCSR1 | 16 | H'FFFFD002 | | 8, 16 | B: 2 | |
| Input level control/status register 2 | ICSR2 | 16 | H'FFFFD004 | | 8, 16, 32 | W: 2 | |
| Output level control/status register 2 | OCSR2 | 16 | H'FFFFD006 | | 8, 16 | L: 4 | |
| Input level control/status register 3 | ICSR3 | 16 | H'FFFFD008 | | 8, 16 | | |
| Software port output enable register | SPOER | 8 | H'FFFFD00A | | 8 | | |
| Port output enable control register 1 | POECR1 | 8 | H'FFFFD00B | | 8 | | |
| Port output enable control register 2 | POECR2 | 16 | H'FFFFD00C | | 8, 16 | | |
| Port A data register L | PADRL | 16 | H'FFFFD102 | I/O | 8, 16 | Pϕ (reference clock) | 16 bits |
| Port A I/O register L | PAIORL | 16 | H'FFFFD106 | PFC | 8, 16 | B: 2 | |
| Port A control register L4 | PACRL4 | 16 | H'FFFFD110 | | 8, 16, 32 | W: 2 | |
| Port A control register L3 | PACRL3 | 16 | H'FFFFD112 | | 8, 16 | L: 4 | |
| Port A control register L2 | PACRL2 | 16 | H'FFFFD114 | | 8, 16, 32 | | |
| Port A control register L1 | PACRL1 | 16 | H'FFFFD116 | | 8, 16 | | |
| Port A port register L | PAPRL | 16 | H'FFFFD11E | I/O | 8, 16 | | |
| Port B data register L | PBDRL | 16 | H'FFFFD182 | | 8, 16 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Port B I/O register L | PBIORL | 16 | H'FFFFD186 | PFC | 8, 16 | Pϕ (reference clock) | 16 bits |
| Port B control register L2 | PBCRL2 | 16 | H'FFFFD194 | | 8, 16, 32 | B: 2 | |
| Port B control register L1 | PBCRL1 | 16 | H'FFFFD196 | | 8, 16 | W: 2 | |
| Port B port register L | PBPRL | 16 | H'FFFFD19E | I/O | 8, 16 | L: 4 | |
| Port D data register L | PDDRL | 16 | H'FFFFD282 | | 8, 16 | | |
| Port D I/O register L | PDIORL | 16 | H'FFFFD286 | PFC | 8, 16 | | |
| Port D control register L3 | PDCRL3 | 16 | H'FFFFD292 | | 8, 16 | | |
| Port D control register L2 | PDCRL2 | 16 | H'FFFFD294 | | 8, 16, 32 | | |
| Port D control register L1 | PDCRL1 | 16 | H'FFFFD296 | | 8, 16 | | |
| Port D port register L | PDPRL | 16 | H'FFFFD29E | I/O | 8, 16 | | |
| Port E data register H | PEDRH | 16 | H'FFFFD300 | | 8, 16, 32 | | |
| Port E data register L | PEDRL | 16 | H'FFFFD302 | | 8, 16 | | |
| Port E I/O register H | PEIORH | 16 | H'FFFFD304 | PFC | 8, 16, 32 | | |
| Port E I/O register L | PEIORL | 16 | H'FFFFD306 | | 8, 16 | | |
| Port E control register H2 | PECRH2 | 16 | H'FFFFD30C | | 8, 16, 32 | | |
| Port E control register H1 | PECRH1 | 16 | H'FFFFD30E | | 8, 16 | | |
| Port E control register L4 | PECRL4 | 16 | H'FFFFD310 | | 8, 16, 32 | | |
| Port E control register L3 | PECRL3 | 16 | H'FFFFD312 | | 8, 16 | | |
| Port E control register L2 | PECRL2 | 16 | H'FFFFD314 | | 8, 16, 32 | | |
| Port E control register L1 | PECRL1 | 16 | H'FFFFD316 | | 8, 16 | | |
| Port E port register H | PEPRH | 16 | H'FFFFD31C | I/O | 8, 16, 32 | | |
| Port E port register L | PEPRL | 16 | H'FFFFD31E | | 8, 16 | | |
| IRQOUT function control register | IFCR | 16 | H'FFFFD322 | PFC | 8, 16 | | |
| Port F data register L | PFDRL | 16 | H'FFFFD382 | I/O | 8, 16 | | |
| A/D control register_0 | ADCR_0 | 8 | H'FFFFD400 | A/D | 8 | Pϕ (reference clock) | 16 bits |
| A/D status register_0 | ADSR_0 | 8 | H'FFFFD402 | (Channel 0) | 8 | B: 2 | |
| A/D start trigger select register_0 | ADSTRGR_0 | 8 | H'FFFFD41C | | 8 | W: 2 | |
| A/D analog input channel select register_0 | ADANSR_0 | 8 | H'FFFFD420 | | 8 | | |
| A/D data register 0 | ADDR0 | 16 | H'FFFFD440 | | 16 | | |
| A/D data register 1 | ADDR1 | 16 | H'FFFFD442 | | 16 | | |
| A/D data register 2 | ADDR2 | 16 | H'FFFFD444 | | 16 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| A/D data register 3 | ADDR3 | 16 | H'FFFFD446 | A/D (Channel 0) | 16 | Pϕ (reference clock) B: 2 W: 2 | 16 bits |
| A/D data register 4 | ADDR4 | 16 | H'FFFFD448 | | 16 | | |
| A/D data register 5 | ADDR5 | 16 | H'FFFFD44A | | 16 | | |
| A/D data register 6 | ADDR6 | 16 | H'FFFFD44C | | 16 | | |
| A/D data register 7 | ADDR7 | 16 | H'FFFFD44E | | 16 | | |
| A/D control register_1 | ADCR_1 | 8 | H'FFFFD600 | A/D (Channel 1) | 8 | Pϕ (reference clock) B: 2 W: 2 | 16 bits |
| A/D status register_1 | ADSR_1 | 8 | H'FFFFD602 | | 8 | | |
| A/D start trigger select register_1 | ADSTRGR_1 | 8 | H'FFFFD61C | | 8 | | |
| A/D analog input channel select register_1 | ADANSR_1 | 8 | H'FFFFD620 | | 8 | | |
| A/D data register 8 | ADDR8 | 16 | H'FFFFD640 | | 16 | | |
| A/D data register 9 | ADDR9 | 16 | H'FFFFD642 | | 16 | | |
| A/D data register 10 | ADDR10 | 16 | H'FFFFD644 | | 16 | | |
| A/D data register 11 | ADDR11 | 16 | H'FFFFD646 | | 16 | | |
| A/D data register 12 | ADDR12 | 16 | H'FFFFD648 | | 16 | | |
| A/D data register 13 | ADDR13 | 16 | H'FFFFD64A | | 16 | | |
| A/D data register 14 | ADDR14 | 16 | H'FFFFD64C | | 16 | | |
| A/D data register 15 | ADDR15 | 16 | H'FFFFD64E | | 16 | | |
| Master control register_0 | MCR | 16 | H'FFFFD800 | RCAN-ET | 16 | Pϕ (reference clock) B: 2 W: 2 L: 4 | 16 bits |
| General status register_0 | GSR | 16 | H'FFFFD802 | | 16 | | |
| Bit configuration register 1_0 | BCR1 | 16 | H'FFFFD804 | | 16 | | |
| Bit configuration register 0_0 | BCR0 | 16 | H'FFFFD806 | | 16 | | |
| Interrupt request register_0 | IRR | 16 | H'FFFFD808 | | 16 | | |
| Interrupt mask register_0 | IMR_0 | 16 | H'FFFFD80A | | 16 | | |
| Transmit error counter/ Receive error counter | TEC_0/REC_0 | 16 | H'FFFFD80C | | 16 | | |
| Transmit wait register 1, transmit wait register 0 | TXPR1_0, TXPR0_0 | 32 | H'FFFFD820 | | 32 | | |
| Transmit cancel register 0 | TXCR0_0 | 16 | H'FFFFD82A | | 16 | | |
| Transmit acknowledge register 0 | TXACK0_0 | 16 | H'FFFFD832 | | 16 | | |
| Abort acknowledge register 0 | ABACK0_0 | 16 | H'FFFFD83A | | 16 | | |
| Receive end register 0 | RXPR0_0 | 16 | H'FFFFD842 | | 16 | | |
| Remote frame request register 0 | RFPR0_0 | 16 | H'FFFFD84A | | 16 | | |

| Register Name | | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|---|
| Mailbox interrupt mask register 0 | | MBIMR0 | 16 | H'FFFFD852 | RCAN-ET | 16 | Pϕ (reference clock)<br>B: 2<br>W: 2<br>L: 4 | 16 bits |
| Unread message status register 0 | | UMSR0 | 16 | H'FFFFD85A | | 16 | | |
| MB[0]. | CONTROL0H | — | 16 | H'FFFFD900 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFD902 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFD904 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFD906 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFD908 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFD909 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFD90A | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFD90B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFD90C | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFD90D | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFD90E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFD90F | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFD910 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFD911 | | 8 | | |
| MB[1]. | CONTROL0H | — | 16 | H'FFFFD920 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFD922 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFD924 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFD926 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFD928 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFD929 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFD92A | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFD92B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFD92C | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFD92D | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFD92E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFD92F | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFD930 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFD931 | | 8 | | |

| Register Name | | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|---|
| MB[2]. | CONTROL0H | — | 16 | H'FFFFD940 | RCAN-ET | 16, 32 | Pϕ (reference clock) | 16 bits |
| | CONTROL0L | — | 16 | H'FFFFD942 | | 16 | B: 2 | |
| | LAFMH | — | 16 | H'FFFFD944 | | 16, 32 | W: 2 | |
| | LAFML | — | 16 | H'FFFFD946 | | 16 | L: 4 | |
| | MSG_DATA[0] | — | 8 | H'FFFFD948 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFD949 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFD94A | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFD94B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFD94C | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFD94D | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFD94E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFD94F | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFD950 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFD951 | | 8 | | |
| MB[3]. | CONTROL0H | — | 16 | H'FFFFD960 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFD962 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFD964 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFD966 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFD968 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFD969 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFD96A | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFD96B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFD96C | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFD96D | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFD96E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFD96F | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFD970 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFD971 | | 8 | | |
| MB[4]. | CONTROL0H | — | 16 | H'FFFFD980 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFD982 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFD984 | | 16, 32 | | |

| Register Name | | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|---|
| MB[4]. | LAFML | — | 16 | H'FFFFD986 | RCAN-ET | 16 | Pϕ (reference clock) | 16 bits |
| | MSG_DATA[0] | — | 8 | H'FFFFD988 | | 8, 16, 32 | B: 2 | |
| | MSG_DATA[1] | — | 8 | H'FFFFD989 | | 8 | W: 2 | |
| | MSG_DATA[2] | — | 8 | H'FFFFD98A | | 8, 16 | L: 4 | |
| | MSG_DATA[3] | — | 8 | H'FFFFD98B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFD98C | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFD98D | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFD98E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFD98F | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFD990 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFD991 | | 8 | | |
| MB[5]. | CONTROL0H | — | 16 | H'FFFFD9A0 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFD9A2 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFD9A4 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFD9A6 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFD9A8 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFD9A9 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFD9AA | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFD9AB | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFD9AC | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFD9AD | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFD9AE | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFD9AF | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFD9B0 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFD9B1 | | 8 | | |
| MB[6]. | CONTROL0H | — | 16 | H'FFFFD9C0 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFD9C2 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFD9C4 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFD9C6 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFD9C8 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFD9C9 | | 8 | | |

| Register Name | | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|---|
| MB[6]. | MSG_DATA[2] | — | 8 | H'FFFFD9CA | RCAN-ET | 8, 16 | Pϕ (reference clock) | 16 bits |
| | MSG_DATA[3] | — | 8 | H'FFFFD9CB | (channel 0) | 8 | B: 2 | |
| | MSG_DATA[4] | — | 8 | H'FFFFD9CC | | 8, 16, 32 | W: 2 | |
| | MSG_DATA[5] | — | 8 | H'FFFFD9CD | | 8 | L: 4 | |
| | MSG_DATA[6] | — | 8 | H'FFFFD9CE | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFD9CF | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFD9D0 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFD9D1 | | 8 | | |
| MB[7]. | CONTROL0H | — | 16 | H'FFFFD9E0 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFD9E2 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFD9E4 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFD9E6 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFD9E8 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFD9E9 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFD9EA | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFD9EB | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFD9EC | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFD9ED | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFD9EE | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFD9EF | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFD9F0 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFD9F1 | | 8 | | |
| MB[8]. | CONTROL0H | — | 16 | H'FFFFDA00 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFDA02 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFDA04 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFDA06 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFDA08 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFDA09 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFDA0A | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFDA0B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFDA0C | | 8, 16, 32 | | |

| Register Name | | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|---|
| MB[8]. | MSG_DATA[5] | — | 8 | H'FFFFDA0D | RCAN-ET (channel 0) | 8 | Pϕ (reference clock) B: 2 W: 2 L: 4 | 16 bits |
| | MSG_DATA[6] | — | 8 | H'FFFFDA0E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFDA0F | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFDA10 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFDA11 | | 8 | | |
| MB[9]. | CONTROL0H | — | 16 | H'FFFFDA20 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFDA22 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFDA24 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFDA26 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFDA28 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFDA29 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFDA2A | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFDA2B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFDA2C | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFDA2D | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFDA2E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFDA2F | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFDA30 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFDA31 | | 8 | | |
| MB[10]. | CONTROL0H | — | 16 | H'FFFFDA40 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFDA42 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFDA44 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFDA46 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFDA48 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFDA49 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFDA4A | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFDA4B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFDA4C | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFDA4D | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFDA4E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFDA4F | | 8 | | |

| Register Name | | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|---|
| MB[10]. | CONTROL1H | — | 8 | H'FFFFDA50 | RCAN-ET (channel 0) | 8, 16 | Pϕ (reference clock) B: 2 W: 2 L: 4 | 16 bits |
| | CONTROL1L | — | 8 | H'FFFFDA51 | | 8 | | |
| MB[11]. | CONTROL0H | — | 16 | H'FFFFDA60 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFDA62 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFDA64 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFDA66 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFDA68 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFDA69 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFDA6A | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFDA6B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFDA6C | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFDA6D | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFDA6E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFDA6F | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFDA70 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFDA71 | | 8 | | |
| MB[12]. | CONTROL0H | — | 16 | H'FFFFDA80 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFDA82 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFDA84 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFDA86 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFDA88 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFDA89 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFDA8A | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFDA8B | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFDA8C | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFDA8D | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFDA8E | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFDA8F | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFDA90 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFDA91 | | 8 | | |

| Register Name | | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|---|
| MB[13]. | CONTROL0H | — | 16 | H'FFFFDAA0 | RCAN-ET (channel 0) | 16, 32 | Pϕ (reference clock) B: 2 W: 2 L: 4 | 16 bits |
| | CONTROL0L | — | 16 | H'FFFFDAA2 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFDAA4 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFDAA6 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFDAA8 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFDAA9 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFDAAA | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFDAAB | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFDAAC | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFDAAD | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFDAAE | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFDAAF | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFDAB0 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFDAB1 | | 8 | | |
| MB[14]. | CONTROL0H | — | 16 | H'FFFFDAC0 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFDAC2 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFDAC4 | | 16, 32 | | |
| | LAFML | — | 16 | H'FFFFDAC6 | | 16 | | |
| | MSG_DATA[0] | — | 8 | H'FFFFDAC8 | | 8, 16, 32 | | |
| | MSG_DATA[1] | — | 8 | H'FFFFDAC9 | | 8 | | |
| | MSG_DATA[2] | — | 8 | H'FFFFDACA | | 8, 16 | | |
| | MSG_DATA[3] | — | 8 | H'FFFFDACB | | 8 | | |
| | MSG_DATA[4] | — | 8 | H'FFFFDACC | | 8, 16, 32 | | |
| | MSG_DATA[5] | — | 8 | H'FFFFDACD | | 8 | | |
| | MSG_DATA[6] | — | 8 | H'FFFFDACE | | 8, 16 | | |
| | MSG_DATA[7] | — | 8 | H'FFFFDACF | | 8 | | |
| | CONTROL1H | — | 8 | H'FFFFDAD0 | | 8, 16 | | |
| | CONTROL1L | — | 8 | H'FFFFDAD1 | | 8 | | |
| MB[15]. | CONTROL0H | — | 16 | H'FFFFDAE0 | | 16, 32 | | |
| | CONTROL0L | — | 16 | H'FFFFDAE2 | | 16 | | |
| | LAFMH | — | 16 | H'FFFFDAE4 | | 16, 32 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| MB[15]. LAFML | — | 16 | H'FFFFDAE6 | RCAN-ET (channel 0) | 16 | Pϕ (reference clock) B: 2 W: 2 L: 4 | 16 bits |
| MSG_DATA[0] | — | 8 | H'FFFFDAE8 | | 8, 16, 32 | | |
| MSG_DATA[1] | — | 8 | H'FFFFDAE9 | | 8 | | |
| MSG_DATA[2] | — | 8 | H'FFFFDAEA | | 8, 16 | | |
| MSG_DATA[3] | — | 8 | H'FFFFDAEB | | 8 | | |
| MSG_DATA[4] | — | 8 | H'FFFFDAEC | | 8, 16, 32 | | |
| MSG_DATA[5] | — | 8 | H'FFFFDAED | | 8 | | |
| MSG_DATA[6] | — | 8 | H'FFFFDAEE | | 8, 16 | | |
| MSG_DATA[7] | — | 8 | H'FFFFDAEF | | 8 | | |
| CONTROL1H | — | 8 | H'FFFFDAF0 | | 8, 16 | | |
| CONTROL1L | — | 8 | H'FFFFDAF1 | | 8 | | |
| Frequency control register | FRQCR | 16 | H'FFFFE800 | CPG | 16 | Pϕ (reference clock) W: 2 | 16 bits |
| Standby control register 1 | STBCR1 | 8 | H'FFFFE802 | Power-down modes | 8 | Pϕ (reference clock) B: 2 | 16 bits |
| Standby control register 2 | STBCR2 | 8 | H'FFFFE804 | | 8 | | |
| Standby control register 3 | STBCR3 | 8 | H'FFFFE806 | | 8 | | |
| Standby control register 4 | STBCR4 | 8 | H'FFFFE808 | | 8 | | |
| Standby control register 5 | STBCR5 | 8 | H'FFFFE80A | | 8 | | |
| Standby control register 6 | STBCR6 | 8 | H'FFFFE80C | | 8 | | |
| Watchdog timer counter | WTCNT | 8 | H'FFFFE810 | WDT *1: Read *2: Write | 8*1, 16*2 | Pϕ (reference clock) B: 2*1 W: 2*2 | 16 bits |
| Watchdog timer control/status register | WTCSR | 8 | H'FFFFE812 | | 8*1, 16*2 | | |
| Oscillation stop detection control register | OSCCR | 8 | H'FFFFE814 | CPG | 8 | Pϕ (reference clock) B: 2 | 16 bits |
| RAM control register | RAMCR | 8 | H'FFFFE880 | Power-down modes | 8 | Pϕ (reference clock) B: 2 | 16 bits |
| Bus function extending register | BSCEHR | 16 | H'FFFFE89A | BSC | 8, 16 | Pϕ (reference clock) B: 2 W: 2 | 16 bits |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Interrupt control register 0 | ICR0 | 16 | H'FFFFE900 | INTC | 8, 16 | Pϕ (reference clock) | 16 bits |
| IRQ control register | IRQCR | 16 | H'FFFFE902 | | 8, 16 | B: 2 | |
| IRQ status register | IRQSR | 16 | H'FFFFE904 | | 8, 16 | W: 2 | |
| Interrupt priority register A | IPRA | 16 | H'FFFFE906 | | 8, 16 | | |
| Interrupt priority register D | IPRD | 16 | H'FFFFE982 | | 16 | | |
| Interrupt priority register E | IPRE | 16 | H'FFFFE984 | | 16 | | |
| Interrupt priority register F | IPRF | 16 | H'FFFFE986 | | 16 | | |
| Interrupt priority register H | IPRH | 16 | H'FFFFE98A | | 16 | | |
| Interrupt priority register I | IPRI | 16 | H'FFFFE98C | | 16 | | |
| Interrupt priority register J | IPRJ | 16 | H'FFFFE98E | | 16 | | |
| Interrupt priority register K | IPRK | 16 | H'FFFFE990 | | 16 | | |
| Interrupt priority register L | IPRL | 16 | H'FFFFE992 | | 16 | | |
| Interrupt priority register M | IPRM | 16 | H'FFFFE994 | | 16 | | |
| Common control register | CMNCR | 32 | H'FFFFF000 | BSC | 32 | Bϕ (reference clock) | 16 bits |
| CS0 space bus control register | CS0BCR | 32 | H'FFFFF004 | | 32 | L: 2 | |
| CS1 space bus control register | CS1BCR | 32 | H'FFFFF008 | | 32 | | |
| CS0 space wait control register | CS0WCR | 32 | H'FFFFF028 | | 32 | | |
| CS1 space wait control register | CS1WCR | 32 | H'FFFFF02C | | 32 | | |
| RAM emulation register | RAMER | 16 | H'FFFFF108 | FLASH | 16 | Bϕ (reference clock)<br>W: 2 | 16 bits |
| Break address register A | BARA | 32 | H'FFFFF300 | UBC | 32 | Bϕ (reference clock) | 16 bits |
| Break address mask register A | BAMRA | 32 | H'FFFFF304 | | 32 | B: 2 | |
| Break bus cycle register A | BBRA | 16 | H'FFFFF308 | | 16 | W: 2<br>L: 2 | |
| Break data register A | BDRA | 32 | H'FFFFF310 | | 32 | | |
| Break data mask register A | BDMRA | 32 | H'FFFFF314 | | 32 | | |
| Break address register B | BARB | 32 | H'FFFFF320 | | 32 | | |
| Break address mask register B | BAMRB | 32 | H'FFFFF324 | | 32 | | |
| Break bus cycle register B | BBRB | 16 | H'FFFFF328 | | 16 | | |
| Break data register B | BDRB | 32 | H'FFFFF330 | | 32 | | |
| Break data mask register B | BDMRB | 32 | H'FFFFF334 | | 32 | | |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Break control register | BRCR | 32 | H'FFFFF3C0 | UBC | 32 | Iϕ (reference clock) | 16 bits |
| Branch source register | BRSR | 32 | H'FFFFF3D0 | | 32 | B: 2 | |
| Branch destination register | BRDR | 32 | H'FFFFF3D4 | | 32 | W: 2 | |
| Execution times break register | BETR | 16 | H'FFFFF3DC | | 16 | L: 2 | |

## 25.2 Register Bit List

Addresses and bit names of each on-chip peripheral module are shown below.

As for 16-bit or 32-bit registers, they are shown in two or four rows.

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| SCSMR_0 | C/$\overline{A}$ | CHR | PE | O/$\overline{E}$ | STOP | MP | CKS[1:0] | | SCI |
| SCBRR_0 | | | | | | | | | (Channel 0) |
| SCSCR_0 | TIE | RIE | TE | RE | MPIE | TEIE | CKE[1:0] | | |
| SCTDR_0 | | | | | | | | | |
| SCSSR_0 | TDRE | RDRF | ORER | FER | PER | TEND | MPB | MPBT | |
| SCRDR_0 | | | | | | | | | |
| SCSDCR_0 | — | — | — | — | DIR | — | — | — | |
| SCSPTR_0 | EIO | — | — | — | SPB1IO | SPB1DT | SPB0IO | SPB0DT | |
| SCSMR_1 | C/$\overline{A}$ | CHR | PE | O/$\overline{E}$ | STOP | MP | CKS[1:0] | | SCI |
| SCBRR_1 | | | | | | | | | (Channel 1) |
| SCSCR_1 | TIE | RIE | TE | RE | MPIE | TEIE | CKE[1:0] | | |
| SCTDR_1 | | | | | | | | | |
| SCSSR_1 | TDRE | RDRF | ORER | FER | PER | TEND | MPB | MPBT | |
| SCRDR_1 | | | | | | | | | |
| SCSDCR_1 | — | — | — | — | DIR | — | — | — | |
| SCSPTR_1 | EIO | — | — | — | SPB1IO | SPB1DT | SPB0IO | SPB0DT | |
| SCSMR_2 | C/$\overline{A}$ | CHR | PE | O/$\overline{E}$ | STOP | MP | CKS[1:0] | | SCI |
| SCBRR_2 | | | | | | | | | (Channel 2) |
| SCSCR_2 | TIE | RIE | TE | RE | MPIE | TEIE | CKE[1:0] | | |
| SCTDR_2 | | | | | | | | | |
| SCSSR_2 | TDRE | RDRF | ORER | FER | PER | TEND | MPB | MPBT | |
| SCRDR_2 | | | | | | | | | |
| SCSDCR_2 | — | — | — | — | DIR | — | — | — | |
| SCSPTR_2 | EIO | — | — | — | SPB1IO | SPB1DT | SPB0IO | SPB0DT | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TCR_3 | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | | MTU2 |
| TCR_4 | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | | |
| TMDR_3 | — | — | BFB | BFA | MD[3:0] | | | | |
| TMDR_4 | — | — | BFB | BFA | MD[3:0] | | | | |
| TIORH_3 | IOB[3:0] | | | | IOA[3:0] | | | | |
| TIORL_3 | IOD[3:0] | | | | IOC[3:0] | | | | |
| TIORH_4 | IOB[3:0] | | | | IOA[3:0] | | | | |
| TIORL_4 | IOD[3:0] | | | | IOC[3:0] | | | | |
| TIER_3 | TTGE | — | — | TCIEV | TGIED | TGIEC | TGIEB | TGIEA | |
| 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 | OLSP | |
| TOCR2 | BF[1:0] | | OLS3N | OLS3P | OLS2N | OLS2P | OLS1N | OLS1P | |
| TCNT_3 | | | | | | | | | |
| | | | | | | | | | |
| TCNT_4 | | | | | | | | | |
| | | | | | | | | | |
| TCDR | | | | | | | | | |
| | | | | | | | | | |
| TDDR | | | | | | | | | |
| | | | | | | | | | |
| TGRA_3 | | | | | | | | | |
| | | | | | | | | | |
| TGRB_3 | | | | | | | | | |
| | | | | | | | | | |
| TGRA_4 | | | | | | | | | |
| | | | | | | | | | |
| TGRB_4 | | | | | | | | | |
| | | | | | | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TCNTS | | | | | | | | | MTU2 |
| | | | | | | | | | |
| 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 | |
| TITCR | T3AEN | 3ACOR[2:0] | | | T4VEN | 4VCOR[2:0] | | | |
| TITCNT | — | 3ACNT[2:0] | | | — | 4VCNT[2:0] | | | |
| TBTER | — | — | — | — | — | — | BTE[1:0] | | |
| TDER | — | — | — | — | — | — | — | TDER | |
| TOLBR | — | — | OLS3N | OLS3P | OLS2N | OLS2P | OLS1N | OLS1P | |
| TBTM_3 | — | — | — | — | — | — | TTSB | TTSA | |
| TBTM_4 | — | — | — | — | — | — | TTSB | TTSA | |
| TADCR | BF[1:0] | | — | — | — | — | — | — | |
| | UT4AE | DT4AE | UT4BE | DT4BE | ITA3AE | ITA4VE | ITB3AE | ITB4VE | |
| TADCORA_4 | | | | | | | | | |
| | | | | | | | | | |
| TADCORB_4 | | | | | | | | | |
| | | | | | | | | | |
| TADCOBRA_4 | | | | | | | | | |
| | | | | | | | | | |
| TADCOBRB_4 | | | | | | | | | |
| | | | | | | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TWCR | CCE | — | — | — | — | — | — | WRE | MTU2 |
| TSTR | CST4 | CST3 | — | — | — | CST2 | CST1 | CST0 | |
| TSYR | SYNC4 | SYNC3 | — | — | — | SYNC2 | SYNC1 | SYNC0 | |
| TCSYSTR | SCH0 | SCH1 | SCH2 | SCH3 | SCH4 | — | SCH3S | SCH4S | |
| TRWER | — | — | — | — | — | — | — | RWE | |
| TCR_0 | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | | |
| TMDR_0 | — | BFE | BFB | BFA | MD[3:0] | | | | |
| TIORH_0 | IOB[3:0] | | | | IOA[3:0] | | | | |
| TIORL_0 | IOD[3:0] | | | | IOC[3:0] | | | | |
| TIER_0 | TTGE | — | — | TCIEV | TGIED | TGIEC | TGIEB | TGIEA | |
| TSR_0 | — | — | — | TCFV | TGFD | TGFC | TGFB | TGFA | |
| TCNT_0 | | | | | | | | | |
| | | | | | | | | | |
| TGRA_0 | | | | | | | | | |
| | | | | | | | | | |
| TGRB_0 | | | | | | | | | |
| | | | | | | | | | |
| TGRC_0 | | | | | | | | | |
| | | | | | | | | | |
| TGRD_0 | | | | | | | | | |
| | | | | | | | | | |
| TGRE_0 | | | | | | | | | |
| | | | | | | | | | |
| TGRF_0 | | | | | | | | | |
| | | | | | | | | | |
| TIER2_0 | TTGE2 | — | — | — | — | — | TGIEF | TGIEE | |
| TSR2_0 | — | — | — | — | — | — | TGFF | TGFE | |
| TBTM_0 | — | — | — | — | — | TTSE | TTSB | TTSA | |
| TCR_1 | — | CCLR[1:0] | | CKEG[1:0] | | TPSC[2:0] | | | |
| TMDR_1 | — | — | — | — | MD[3:0] | | | | |
| TIOR_1 | IOB[3:0] | | | | IOA[3:0] | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TIER_1 | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA | MTU2 |
| TSR_1 | TCFD | — | TCFU | TCFV | — | — | TGFB | TGFA | |
| TCNT_1 | | | | | | | | | |
| | | | | | | | | | |
| TGRA_1 | | | | | | | | | |
| | | | | | | | | | |
| TGRB_1 | | | | | | | | | |
| | | | | | | | | | |
| TICCR | — | — | — | — | I2BE | I2AE | I1BE | I1AE | |
| TCR_2 | — | CCLR[1:0] | | CKEG[1:0] | | TPSC[2:0] | | | |
| TMDR_2 | — | — | — | — | MD[3:0] | | | | |
| TIOR_2 | IOB[3:0] | | | | IOA[3:0] | | | | |
| TIER_2 | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA | |
| TSR_2 | TCFD | — | TCFU | TCFV | — | — | TGFB | TGFA | |
| TCNT_2 | | | | | | | | | |
| | | | | | | | | | |
| TGRA_2 | | | | | | | | | |
| | | | | | | | | | |
| TGRB_2 | | | | | | | | | |
| | | | | | | | | | |
| TCNTU_5 | | | | | | | | | |
| | | | | | | | | | |
| TGRU_5 | | | | | | | | | |
| | | | | | | | | | |
| TCRU_5 | — | — | — | — | — | — | TPSC[1:0] | | |
| TIORU_5 | — | — | — | IOC[4:0] | | | | | |
| TCNTV_5 | | | | | | | | | |
| | | | | | | | | | |
| TGRV_5 | | | | | | | | | |
| | | | | | | | | | |
| TCRV_5 | — | — | — | — | — | — | TPSC[1:0] | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TIORV_5 | — | — | — | IOC[4:0] | | | | | MTU2 |
| TCNTW_5 | | | | | | | | | |
| | | | | | | | | | |
| TGRW_5 | | | | | | | | | |
| | | | | | | | | | |
| TCRW_5 | — | — | — | — | — | — | TPSC[1:0] | | |
| TIORW_5 | — | — | — | IOC[4:0] | | | | | |
| TSR_5 | — | — | — | — | — | CMFU5 | CMFV5 | CMFW5 | |
| TIER_5 | — | — | — | — | — | TGIE5U | TGIE5V | TGIE5W | |
| TSTR_5 | — | — | — | — | — | CSTU5 | CSTV5 | CSTW5 | |
| TCNTCMPCLR | — | — | — | — | — | CMPCLR5U | CMPCLR5V | CMPCLR5W | |
| TCR_3S | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | | MTU2S |
| TCR_4S | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | | |
| TMDR_3S | — | — | BFB | BFA | MD[3:0] | | | | |
| TMDR_4S | — | — | BFB | BFA | MD[3:0] | | | | |
| TIORH_3S | IOB[3:0] | | | | IOA[3:0] | | | | |
| TIORL_3S | IOD[3:0] | | | | IOC[3:0] | | | | |
| TIORH_4S | IOB[3:0] | | | | IOA[3:0] | | | | |
| TIORL_4S | IOD[3:0] | | | | IOC[3:0] | | | | |
| TIER_3S | TTGE | — | — | TCIEV | TGIED | TGIEC | TGIEB | TGIEA | |
| TIER_4S | TTGE | TTGE2 | — | TCIEV | TGIED | TGIEC | TGIEB | TGIEA | |
| TOERS | — | — | OE4D | OE4C | OE3D | OE4B | OE4A | OE3B | |
| TGCRS | — | BDC | N | P | FB | WF | VF | UF | |
| TOCR1S | — | PSYE | — | — | TOCL | TOCS | OLSN | OLSP | |
| TOCR2S | BF[1:0] | | OLS3N | OLS3P | OLS2N | OLS2P | OLS1N | OLS1P | |
| TCNT_3S | | | | | | | | | |
| | | | | | | | | | |
| TCNT_4S | | | | | | | | | |
| | | | | | | | | | |
| TCDRS | | | | | | | | | |
| | | | | | | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TDDRS | | | | | | | | | MTU2S |
| | | | | | | | | | |
| TGRA_3S | | | | | | | | | |
| | | | | | | | | | |
| TGRB_3S | | | | | | | | | |
| | | | | | | | | | |
| TGRA_4S | | | | | | | | | |
| | | | | | | | | | |
| TGRB_4S | | | | | | | | | |
| | | | | | | | | | |
| TCNTSS | | | | | | | | | |
| | | | | | | | | | |
| TCBRS | | | | | | | | | |
| | | | | | | | | | |
| TGRC_3S | | | | | | | | | |
| | | | | | | | | | |
| TGRD_3S | | | | | | | | | |
| | | | | | | | | | |
| TGRC_4S | | | | | | | | | |
| | | | | | | | | | |
| TGRD_4S | | | | | | | | | |
| | | | | | | | | | |
| TSR_3S | TCFD | — | — | TCFV | TGFD | TGFC | TGFB | TGFA | |
| TSR_4S | TCFD | — | — | TCFV | TGFD | TGFC | TGFB | TGFA | |
| TITCRS | T3AEN | 3ACOR[2:0] | | | T4VEN | 4VCOR[2:0] | | | |
| TITCNTS | — | 3ACNT[2:0] | | | — | 4VCNT[2:0] | | | |
| TBTERS | — | — | — | — | — | — | BTE[1:0] | | |
| TDERS | — | — | — | — | — | — | — | TDER | |
| TOLBRS | — | — | OLS3N | OLS3P | OLS2N | OLS2P | OLS1N | OLS1P | |
| TBTM_3S | — | — | — | — | — | — | TTSB | TTSA | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TBTM_4S | — | — | — | — | — | — | TTSB | TTSA | MTU2S |
| TADCRS | BF[1:0] | | — | — | — | — | — | — | |
| | UT4AE | DT4AE | UT4BE | DT4BE | ITA3AE | ITA4VE | ITB3AE | ITB4VE | |
| TADCORA_4S | | | | | | | | | |
| | | | | | | | | | |
| TADCORB_4S | | | | | | | | | |
| | | | | | | | | | |
| TADCOBRA_4S | | | | | | | | | |
| | | | | | | | | | |
| TADCOBRB_4S | | | | | | | | | |
| | | | | | | | | | |
| TSYCRS | CE0A | CE0B | CE0C | CE0D | CE1A | CE1B | CE2A | CE2B | |
| TWCRS | CCE | — | — | — | — | — | SCC | WRE | |
| TSTRS | CST4 | CST3 | — | — | — | CST2 | CST1 | CST0 | |
| TSYRS | SYNC4 | SYNC3 | — | — | — | SYNC2 | SYNC1 | SYNC0 | |
| TRWERS | — | — | — | — | — | — | — | RWE | |
| TCNTU_5S | | | | | | | | | |
| | | | | | | | | | |
| TGRU_5S | | | | | | | | | |
| | | | | | | | | | |
| TCRU_5S | — | — | — | — | — | — | TPSC[1:0] | | |
| TIORU_5S | — | — | — | IOC[4:0] | | | | | |
| TCNTV_5S | | | | | | | | | |
| | | | | | | | | | |
| TGRV_5S | | | | | | | | | |
| | | | | | | | | | |
| TCRV_5S | — | — | — | — | — | — | TPSC[1:0] | | |
| TIORV_5S | — | — | — | IOC[4:0] | | | | | |
| TCNTW_5S | | | | | | | | | |
| | | | | | | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| TGRW_5S | | | | | | | | | MTU2S |
| | | | | | | | | | |
| TCRW_5S | — | — | — | — | — | — | TPSC[1:0] | | |
| TIORW_5S | — | — | — | IOC[4:0] | | | | | |
| TSR_5S | — | — | — | — | — | CMFU5 | CMFV5 | CMFW5 | |
| TIER_5S | — | — | — | — | — | TGIE5U | TGIE5V | TGIE5W | |
| TSTR_5S | — | — | — | — | — | CSTU5 | CSTV5 | CSTW5 | |
| TCNTCMPCLRS | — | — | — | — | — | CMPCLR5U | CMPCLR5V | CMPCLR5W | |
| FCCS | FWE | MAT | — | FLER | — | — | — | SCO | FLASH |
| FPCS | — | — | — | — | — | — | — | PPVS | |
| FECS | — | — | — | — | — | — | — | EPVB | |
| FKEY | K[7:0] | | | | | | | | |
| FMATS | MS7 | MS6 | MS5 | MS4 | MS3 | MS2 | MS1 | MS0 | |
| FTDAR | TDER | TDA[6:0] | | | | | | | |
| DTCERA | DTCERA15 | DTCERA14 | DTCERA13 | DTCERA12 | — | — | — | — | DTC |
| | — | — | — | — | — | — | — | — | |
| DTCERB | DTCERB15 | DTCERB14 | DTCERB13 | DTCERB12 | DTCERB11 | DTCERB10 | DTCERB9 | DTCERB8 | |
| | DTCERB7 | DTCERB6 | DTCERB5 | DTCERB4 | DTCERB3 | DTCERB2 | DTCERB1 | DTCERB0 | |
| DTCERC | DTCERC15 | DTCERE14 | DTCERE13 | DTCERE12 | — | — | — | — | |
| | — | — | — | — | DTCERC3 | DTCERC2 | DTCERC1 | DTCERC0 | |
| DTCERD | DTCERD15 | DTCERD14 | DTCERD13 | DTCERD12 | DTCERD11 | DTCERD10 | DTCERD9 | DTCERD8 | |
| | DTCERD7 | DTCERD6 | — | — | — | DTCERD2 | DTCERD1 | — | |
| DTCERE | DTCERE15 | DTCERE14 | DTCERE13 | DTCERE12 | DTCERE11 | DTCERE10 | — | — | |
| | DTCERE7 | DTCERE6 | DTCERE5 | DTCERE4 | DTCERE3 | — | — | — | |
| DTCCR | — | — | — | RRS | RCHNE | — | — | ERR | |
| DTCVBR | | | | | | | | | |
| | | | | | | | | | |
| | | | | | — | — | — | — | |
| | — | — | — | — | — | — | — | — | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| ICCR1 | ICE | RCVD | MST | TRS | CKS[3'0] | | | | $I^2C2$ |
| ICCR2 | BBSY | SCP | SDAO | SDAOP | SCLO | — | IICRST | — | |
| ICMR | MLS | WAIT | — | — | BCWP | BC[2:0] | | | |
| ICIER | TIE | TEIE | RIE | NAKE | STIE | ACKE | ACKBR | ACKBT | |
| ICSR | TDRE | TEND | RDRF | NACKF | STOP | AL/OVE | AAS | ADZ | |
| SAR | SVA[6:0] | | | | | | | FS | |
| ICDRT | | | | | | | | | |
| ICDRR | | | | | | | | | |
| NF2CYC | — | — | — | — | — | — | — | NF2CYC | |
| SSCRH | MSS | BIDE | — | SOL | SOLP | — | CSS[1:0] | | SSU |
| SSCRL | FCLRM | SSUMS | SRES | — | — | — | DATS[1:0] | | |
| SSMR | MLS | CPOS | CPHS | — | — | CKS[2:0] | | | |
| SSER | TE | RE | — | — | TEIE | TIE | RIE | CEIE | |
| SSSR | — | ORER | — | — | TEND | TDRE | RDRF | CE | |
| SSCR2 | — | — | — | TENDSTS | SCSATS | SSODTS | — | — | |
| SSTDR0 | | | | | | | | | |
| SSTDR1 | | | | | | | | | |
| SSTDR2 | | | | | | | | | |
| SSTDR3 | | | | | | | | | |
| SSRDR0 | | | | | | | | | |
| SSRDR1 | | | | | | | | | |
| SSRDR2 | | | | | | | | | |
| SSRDR3 | | | | | | | | | |
| CMSTR | — | — | — | — | — | — | — | — | CMT |
| | — | — | — | — | — | — | STR1 | STR0 | |
| CMCSR_0 | — | — | — | — | — | — | — | — | |
| | CMF | CMIE | — | — | — | — | CKS[1:0] | | |
| CMCNT_0 | | | | | | | | | |
| | | | | | | | | | |
| CMCOR_0 | | | | | | | | | |
| | | | | | | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| CMCSR_1 | — | — | — | — | — | — | — | — | CMT |
| | CMF | CMIE | — | — | — | — | CKS[1:0] | | |
| CMCNT_1 | | | | | | | | | |
| | | | | | | | | | |
| CMCOR_1 | | | | | | | | | |
| | | | | | | | | | |
| ICSR1 | — | POE2F | POE1F | POE0F | — | — | — | PIE1 | POE |
| | — | — | POE2M[1:0] | | POE1M[1:0] | | POE0M[1:0] | | |
| OCSR1 | OSF1 | — | — | — | — | — | OCE1 | OIE1 | |
| | — | — | — | — | — | — | — | — | |
| ICSR2 | — | POE6F | POE5F | POE4F | — | — | — | PIE2 | |
| | — | — | POE6M[1:0] | | POE5M[1:0] | | POE4M[1:0] | | |
| OCSR2 | OSF2 | — | — | — | — | — | OCE2 | OIE2 | |
| | — | — | — | — | — | — | — | — | |
| ICSR3 | — | — | — | POE8F | — | — | POE8E | PIE3 | |
| | — | — | — | — | — | — | POE8M[1:0] | | |
| SPOER | — | — | — | — | — | MTU2SHIZ | MTU2CH0HIZ | MTU2CH34HIZ | |
| POECR1 | — | — | — | — | MTU2PE3ZE | MTU2PE2ZE | MTU2PE1ZE | MTU2PE0ZE | |
| POECR2 | — | MTU2P1CZE | MTU2P2CZE | MTU2P3CZE | — | MTU2SP1CZE | MTU2SP2CZE | MTU2SP3CZE | |
| | — | — | — | — | — | — | — | — | |
| PADRL | PA15DR | PA14DR | PA13DR | PA12DR | PA11DR | PA10DR | PA9DR | PA8DR | I/O |
| | PA7DR | PA6DR | PA5DR | PA4DR | PA3DR | PA2DR | PA1DR | PA0DR | |
| PAIORL | PA15IOR | PA14IOR | PA13IOR | PA12IOR | PA11IOR | PA10IOR | PA9IOR | PA8IOR | PFC |
| | PA7IOR | PA6IOR | PA5IOR | PA4IOR | PA3IOR | PA2IOR | PA1IOR | PA0IOR | |
| PACRL4 | — | PA15MD2 | PA15MD1 | PA15MD0 | — | PA14MD2 | PA14MD1 | PA14MD0 | |
| | — | PA13MD2 | PA13MD1 | PA13MD0 | — | PA12MD2 | PA12MD1 | PA12MD0 | |
| PACRL3 | — | PA11MD2 | PA11MD1 | PA11MD0 | — | PA10MD2 | PA10MD1 | PA10MD0 | |
| | — | PA9MD2 | PA9MD1 | PA9MD0 | — | PA8MD2 | PA8MD1 | PA8MD0 | |
| PACRL2 | — | PA7MD2 | PA7MD1 | PA7MD0 | — | PA6MD2 | PA6MD1 | PA6MD0 | |
| | — | PA5MD2 | PA5MD1 | PA5MD0 | — | PA4MD2 | PA4MD1 | PA4MD0 | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| PACRL1 | — | PA3MD2 | PA3MD1 | PA3MD0 | — | PA2MD2 | PA2MD1 | PA2MD0 | PFC |
| | — | PA1MD2 | PA1MD1 | PA1MD0 | — | PA0MD2 | PA0MD1 | PA0MD0 | |
| PAPRL | PA15PR | PA14PR | PA13PR | PA12PR | PA11PR | PA10PR | PA9PR | PA8PR | I/O |
| | PA7PR | PA6PR | PA5PR | PA4PR | PA3PR | PA2PR | PA1PR | PA0PR | |
| PBDRL | — | — | — | — | — | — | — | — | |
| | PB7DR | PB6DR | PB5DR | PB4DR | PB3DR | PB2DR | PB1DR | PB0DR | |
| PBIORL | — | — | — | — | — | — | — | — | PFC |
| | PB7IOR | PB6IOR | PB5IOR | PB4IOR | PB3IOR | PB2IOR | PB1IOR | PB0IOR | |
| PBCRL2 | — | PB7MD2 | PB7MD1 | PB7MD0 | — | PB6MD2 | PB6MD1 | PB6MD0 | |
| | — | PB5MD2 | PB5MD1 | PB5MD0 | — | PB4MD2 | PB4MD1 | PB4MD0 | |
| PBCRL1 | — | PB3MD2 | PB3MD1 | PB3MD0 | — | PB2MD2 | PB2MD1 | PB2MD0 | |
| | — | PB1MD2 | PB1MD1 | PB1MD0 | — | PB0MD2 | PB0MD1 | PB0MD0 | |
| PBPRL | — | — | — | — | — | — | — | — | I/O |
| | PB7PR | PB6PR | PB5PR | PB4PR | PB3PR | PB2PR | PB1PR | PB0PR | |
| PDDRL | — | — | — | — | — | PD10DR | PD9DR | PD8DR | |
| | PD7DR | PD6DR | PD5DR | PD4DR | PD3DR | PD2DR | PD1DR | PD0DR | |
| PDIORL | — | — | — | — | — | PD10IOR | PD9IOR | PD8IOR | PFC |
| | PD7IOR | PD6IOR | PD5IOR | PD4IOR | PD3IOR | PD2IOR | PD1IOR | PD0IOR | |
| PDCRL3 | — | — | — | — | — | PD10MD2 | PD10MD1 | PD10MD0 | |
| | — | PD9MD2 | PD9MD1 | PD9MD0 | — | PD8MD2 | PD8MD1 | PD8MD0 | |
| PDCRL2 | — | PD7MD2 | PD7MD1 | PD7MD0 | — | PD6MD2 | PD6MD1 | PD6MD0 | |
| | — | PD5MD2 | PD5MD1 | PD5MD0 | — | PD4MD2 | PD4MD1 | PD4MD0 | |
| PDCRL1 | — | PD3MD2 | PD3MD1 | PD3MD0 | — | PD2MD2 | PD2MD1 | PD2MD0 | |
| | — | PD1MD2 | PD1MD1 | PD1MD0 | — | PD0MD2 | PD0MD1 | PD0MD0 | |
| PDPRL | — | — | — | — | — | PD10PR | PD9PR | PD8PR | I/O |
| | PD7PR | PD6PR | PD5PR | PD4PR | PD3PR | PD2PR | PD1PR | PD0PR | |
| PEDRH | — | — | — | — | — | — | — | — | |
| | — | — | PE21DR | PE20DR | PE19DR | PE18DR | PE17DR | PE16DR | |
| PEDRL | PE15DR | PE14DR | PE13DR | PE12DR | PE11DR | PE10DR | PE9DR | PE8DR | |
| | PE7DR | PE6DR | PE5DR | PE4DR | PE3DR | PE2DR | PE1DR | PE0DR | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| PEIORH | — | — | — | — | — | — | — | — | PFC |
| | — | — | PE21IOR | PE20IOR | PE19IOR | PE18IOR | PE17IOR | PE16IOR | |
| PEIORL | PE15IOR | PE14IOR | PE13IOR | PE12IOR | PE11IOR | PE10IOR | PE9IOR | PE8IOR | |
| | PE7IOR | PE6IOR | PE5IOR | PE4IOR | PE3IOR | PE2IOR | PE1IOR | PE0IOR | |
| PECRH2 | — | — | — | — | — | — | — | — | |
| | — | — | PE21MD1 | PE21MD0 | — | — | PE20MD1 | PE20MD0 | |
| PECRH1 | — | — | PE19MD1 | PE19MD0 | — | — | PE18MD1 | PE18MD0 | |
| | — | — | PE17MD1 | PE17MD0 | — | PE16MD2 | PE16MD1 | PE16MD0 | |
| PECRL4 | — | PE15MD2 | PE15MD1 | PE15MD0 | — | PE14MD2 | PE14MD1 | PE14MD0 | |
| | — | — | PE13MD1 | PE13MD0 | — | PE12MD2 | PE12MD1 | PE12MD0 | |
| PECRL3 | — | PE11MD2 | PE11MD1 | PE11MD0 | — | PE10MD2 | PE10MD1 | PE10MD0 | |
| | — | PE9MD2 | PE9MD1 | PE9MD0 | — | PE8MD2 | PE8MD1 | PE8MD0 | |
| PECRL2 | — | PE7MD2 | PE7MD1 | PE7MD0 | — | PE6MD2 | PE6MD1 | PE6MD0 | |
| | — | PE5MD2 | PE5MD1 | PE5MD0 | — | PE4MD2 | PE4MD1 | PE4MD0 | |
| PECRL1 | — | PE3MD2 | PE3MD1 | PE3MD0 | — | PE2MD2 | PE2MD1 | PE2MD0 | |
| | — | PE1MD2 | PE1MD1 | PE1MD0 | — | — | PE0MD1 | PE0MD0 | |
| PEPRH | — | — | — | — | — | — | — | — | I/O |
| | — | — | PE21PR | PE20PR | PE19PR | PE18PR | PE17PR | PE16PR | |
| PEPRL | PE15PR | PE14PR | PE13PR | PE12PR | PE11PR | PE10PR | PE9PR | PE8PR | |
| | PE7PR | PE6PR | PE5PR | PE4PR | PE3PR | PE2PR | PE1PR | PE0PR | |
| IFCR | — | — | — | — | — | — | — | — | PFC |
| | — | — | — | — | — | — | IRQMD1 | IRQMD0 | |
| PFDRL | PF15DR | PF14DR | PF13DR | PF12DR | PF11DR | PF10DR | PF9DR | PF8DR | I/O |
| | PF7DR | PF6DR | PF5DR | PF4DR | PF3DR | PF2DR | PF1DR | PF0DR | |
| ADCR_0 | ADST | ADCS | ACE | ADIE | — | — | TRGE | EXTRG | A/D (Channel 0) |
| ADSR_0 | — | — | — | — | — | — | — | ADF | |
| ADSTRGR_0 | — | STR6 | STR5 | STR4 | STR3 | STR2 | STR1 | STR0 | |
| ADANSR_0 | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | |
| ADDR0 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR1 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| ADDR2 | — | — | — | — | ADD[11:8] | | | | A/D (Channel 0) |
| | ADD[7:0] | | | | | | | | |
| ADDR3 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR4 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR5 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR6 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR7 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADCR_1 | ADST | ADCS | ACE | ADIE | — | — | TRGE | EXTRG | A/D (Channel 1) |
| ADSR_1 | — | — | — | — | — | — | — | ADF | |
| ADSTRGR_1 | — | STR6 | STR5 | STR4 | STR3 | STR2 | STR1 | STR0 | |
| ADANSR_1 | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | |
| ADDR8 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR9 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR10 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR11 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR12 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR13 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR14 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |
| ADDR15 | — | — | — | — | ADD[11:8] | | | | |
| | ADD[7:0] | | | | | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| MCR | MCR15 | MCR14 | — | — | — | TST[2:0] | | | RCAN-ET |
| | MCR7 | MCR6 | MCR5 | — | — | MCR2 | MCR1 | MCR0 | |
| GSR | — | — | — | — | — | — | — | — | |
| | — | — | GSR5 | GSR4 | GSR3 | GSR2 | GSR1 | GSR0 | |
| BCR1 | TSG1[3:0] | | | | — | TSG2[2:0] | | | |
| | — | — | SJW[1:0] | | — | — | — | BSP | |
| BCR0 | — | — | — | — | — | — | — | — | |
| | BRP[7:0] | | | | | | | | |
| IRR | — | — | IRR13 | IRR12 | — | — | IRR9 | IRR8 | |
| | IRR7 | IRR6 | IRR5 | IRR4 | IRR3 | IRR2 | IRR1 | IRR0 | |
| IMR | IMR15 | IMR14 | IMR13 | IMR12 | IMR11 | IMR10 | IMR9 | IMR8 | |
| | IMR7 | IMR6 | IMR5 | IMR4 | IMR3 | IMR2 | IMR1 | IMR0 | |
| TEC/REC | TEC7 | TEC6 | TEC5 | TEC4 | TEC3 | TEC2 | TEC1 | TEC0 | |
| | REC7 | REC6 | REC5 | REC4 | REC3 | REC2 | REC1 | REC0 | |
| TXPR1, TXPR 0 | TXPR1[15:8] | | | | | | | | |
| | TXPR1[7:0] | | | | | | | | |
| | TXPR0[15:8] | | | | | | | | |
| | TXPR0[7:1] | | | | | | | — | |
| TXCR0_0 | TXCR0[15:8] | | | | | | | | |
| | TXCR0[7:1] | | | | | | | — | |
| TXACK0 | TXACK0[15:8] | | | | | | | | |
| | TXACK0[7:1] | | | | | | | — | |
| ABACK0 | ABACK0[15:8] | | | | | | | | |
| | ABACK0[7:1] | | | | | | | — | |
| RXPR0_0 | RXPR0[15:8] | | | | | | | | |
| | RXPR0[7:0] | | | | | | | | |
| RFPR0 | RFPR0[15:8] | | | | | | | | |
| | RFPR0[7:0] | | | | | | | | |
| MBIMR0 | MBIMR0[15:8] | | | | | | | | |
| | MBIMR0[7:0] | | | | | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| UMSR0 | UMSR0[15:8] | | | | | | | | RCAN-ET |
| | UMSR0[7:0] | | | | | | | | |
| MB[0]. CONTROL0H | IDE | RTR | — | STDID[10:6] | | | | | RCAN-ET (MCR15 = 1) |
| | STDID[5:0] | | | | | | EXTID[17:16] | | |
| MB[0]. CONTROL0H | — | STDID[10:4] | | | | | | | RCAN-ET (MCR15 = 0) |
| | STDID[3:0] | | | | RTR | IDE | EXTID[17:16] | | |
| MB[0]. CONTROL0L | EXTID[15:8] | | | | | | | | RCAN-ET |
| | EXTID[7:0] | | | | | | | | |
| MB[0]. LAFMH | IDE_LAFM | — | — | STDID_LAFM[10:6] | | | | | RCAN-ET (MCR15 = 1) |
| | STDID_LAFM[5:0] | | | | | | EXTID_LAFM[17:16] | | |
| MB[0]. LAFMH | — | STDID_LAFM[10:4] | | | | | | | RCAN-ET (MCR15 = 0) |
| | STDID_LAFM[3:0] | | | | — | IDE_LAFM | EXTID_LAFM[17:16] | | |
| MB[0]. LAFML | EXTID_LAFM[15:8] | | | | | | | | RCAN-ET |
| | EXTID_LAFM[7:0] | | | | | | | | |
| MB[0]. MSG_DATA[0] | MSG_DATA_0 | | | | | | | | |
| MB[0]. MSG_DATA[1] | MSG_DATA_1 | | | | | | | | |
| MB[0]. MSG_DATA[2] | MSG_DATA_2 | | | | | | | | |
| MB[0]. MSG_DATA[3] | MSG_DATA_3 | | | | | | | | |
| MB[0]. MSG_DATA[4] | MSG_DATA_4 | | | | | | | | |
| MB[0]. MSG_DATA[5] | MSG_DATA_5 | | | | | | | | |
| MB[0]. MSG_DATA[6] | MSG_DATA_6 | | | | | | | | |
| MB[0]. MSG_DATA[7] | MSG_DATA_7 | | | | | | | | |
| MB[0]. CONTROL1H | — | — | NMC | — | — | MBC[2:0] | | | |
| MB[0]. CONTROL1L | — | — | — | — | DLC[3:0] | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| MB[1] | Same bit configuration as MB[0] | | | | | | | | RCAN-ET |
| MB[2] | Same bit configuration as MB[0] | | | | | | | | |
| MB[3] | Same bit configuration as MB[0] | | | | | | | | |
| ↓ | (Repeat) | | | | | | | | |
| MB[13] | Same bit configuration as MB[0] | | | | | | | | |
| MB[14] | Same bit configuration as MB[0] | | | | | | | | |
| MB[15] | Same bit configuration as MB[0] | | | | | | | | |
| FRQCR | — | IFC[2:0] | | | BFC[2:0] | | | PFC[2] | CPG |
| | PFC[1:0] | | MIFC[2:0] | | | MPFC[2:0] | | | |
| STBCR1 | STBY | — | — | — | — | — | — | — | Power-down modes |
| STBCR2 | MSTP7 | MSTP6 | — | MSTP4 | — | — | — | — | |
| STBCR3 | MSTP15 | — | MSTP13 | MSTP12 | MSTP11 | MSTP10 | MSTP9* | MSTP8 | |
| STBCR4 | MSTP23 | MSTP22 | MSTP21 | MSTP20 | MSTP19 | — | — | — | |
| STBCR5 | — | — | — | — | — | — | MSTP25 | MSTP24 | |
| STBCR6 | AUDSRST | HIZ | — | — | — | — | STBYMD | — | |
| WTCNT | | | | | | | | | WDT |
| WTCSR | TME | WT/IT | RSTS | WOVF | IOVF | CKS[2:0] | | | |
| OSCCR | — | — | — | — | — | OSCSTOP | — | OSCERS | CPG |
| RAMCR | — | — | — | RAME | — | — | — | — | Power-down modes |
| BSCEHR | DTLOCK | CSSTP1 | — | CSSTP2 | DTBST | DTSA | CSSTP3 | DTPR | BSC |
| | — | — | — | — | — | — | — | — | |
| ICR0 | NMIL | — | — | — | — | — | — | NMIE | INTC |
| | — | — | — | — | — | — | — | — | |
| IRQCR | — | — | — | — | — | — | — | — | |
| | IRQ31S | IRQ30S | IRQ21S | IRQ20S | IRQ11S | IRQ10S | IRQ01S | IRQ00S | |
| IRQSR | — | — | — | — | IRQ3L | IRQ2L | IRQ1L | IRQ0L | |
| | — | — | — | — | IRQ3F | IRQ2F | IRQ1F | IRQ0F | |
| IPRA | IRQ0 | IRQ0 | IRQ0 | IRQ0 | IRQ1 | IRQ1 | IRQ1 | IRQ1 | |
| | IRQ2 | IRQ2 | IRQ2 | IRQ2 | IRQ3 | IRQ3 | IRQ3 | IRQ3 | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| IPRD | MTU2_0 | MTU2_0 | MTU2_0 | MTU2_0 | MTU2_0 | MTU2_0 | MTU2_0 | MTU2_0 | INTC |
| | MTU2_1 | MTU2_1 | MTU2_1 | MTU2_1 | MTU2_1 | MTU2_1 | MTU2_1 | MTU2_1 | |
| IPRE | MTU2_2 | MTU2_2 | MTU2_2 | MTU2_2 | MTU2_2 | MTU2_2 | MTU2_2 | MTU2_2 | |
| | MTU2_3 | MTU2_3 | MTU2_3 | MTU2_3 | MTU2_3 | MTU2_3 | MTU2_3 | MTU2_3 | |
| IPRF | MTU2_4 | MTU2_4 | MTU2_4 | MTU2_4 | MTU2_4 | MTU2_4 | MTU2_4 | MTU2_4 | |
| | MTU2_5 | MTU2_5 | MTU2_5 | MTU2_5 | POE(MTU2) | POE(MTU2) | POE(MTU2) | POE(MTU2) | |
| IPRH | — | — | — | — | I²C2 | I²C2 | I²C2 | I²C2 | |
| | MTU2S_3 | MTU2S_3 | MTU2S_3 | MTU2S_3 | MTU2S_3 | MTU2S_3 | MTU2S_3 | MTU2S_3 | |
| IPRI | MTU2S_4 | MTU2S_4 | MTU2S_4 | MTU2S_4 | MTU2S_4 | MTU2S_4 | MTU2S_4 | MTU2S_4 | |
| | MTU2S_5 | MTU2S_5 | MTU2S_5 | MTU2S_5 | POE(MTU2S) | POE(MTU2S) | POE(MTU2S) | POE(MTU2S) | |
| IPRJ | CMT_0 | CMT_0 | CMT_0 | CMT_0 | CMT_1 | CMT_1 | CMT_1 | CMT_1 | |
| | — | — | — | — | WDT | WDT | WDT | WDT | |
| IPRK | — | — | — | — | — | — | — | — | |
| | A/D_0 | A/D_0 | A/D_0 | A/D_0 | A/D_1 | A/D_1 | A/D_1 | A/D_1 | |
| IPRL | SCI_0 | SCI_0 | SCI_0 | SCI_0 | SCI_1 | SCI_1 | SCI_1 | SCI_1 | |
| | SCI_2 | SCI_2 | SCI_2 | SCI_2 | — | — | — | — | |
| IPRM | SSU | SSU | SSU | SSU | I²C2 | I²C2 | I²C2 | I²C2 | |
| | RCAN-ET_0 | RCAN-ET_0 | RCAN-ET_0 | RCAN-ET_0 | — | — | — | — | |
| CMNCR | — | — | — | — | — | — | — | — | BSC |
| | — | — | — | — | — | — | — | — | |
| | — | — | — | — | — | — | — | — | |
| | — | — | — | — | — | — | HIZMEM | — | |
| CS0BCR | — | — | IWW[1:0] | | — | IWRWD[1:0] | | — | |
| | IWRWS[1:0] | | — | IWRRD[1:0] | | — | IWRRS[1:0] | | |
| | — | — | — | — | — | BSZ[1:0] | | — | |
| | — | — | — | — | — | — | — | — | |
| CS1BCR | — | — | IWW[1:0] | | — | IWRWD[1:0] | | — | |
| | IWRWS[1:0] | | — | IWRRD[1:0] | | — | IWRRS[1:0] | | |
| | — | — | — | — | — | BSZ[1:0] | | — | |
| | — | — | — | — | — | — | — | — | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| CS0WCR | — | — | — | — | — | — | — | — | BSC |
| | — | — | — | — | — | WW[2:0] | | | |
| | — | — | — | SW[1:0] | | WR[3:1] | | | |
| | WR[0] | WM | — | — | — | — | HW[1:0] | | |
| CS1WCR | — | — | — | — | — | — | — | — | |
| | — | — | — | — | — | WW[2:0] | | | |
| | — | — | — | SW[1:0] | | WR[3:1] | | | |
| | WR[0] | WM | — | — | — | — | HW[1:0] | | |
| RAMER | — | — | — | — | — | — | — | — | FLASH |
| | — | — | — | — | RAMS | RAM[2:0] | | | |
| BARA | BAA31 | BAA30 | BAA29 | BAA28 | BAA27 | BAA26 | BAA25 | BAA24 | UBC |
| | BAA23 | BAA22 | BAA21 | BAA20 | BAA19 | BAA18 | BAA17 | BAA16 | |
| | BAA15 | BAA14 | BAA13 | BAA12 | BAA11 | BAA10 | BAA9 | BAA8 | |
| | BAA7 | BAA6 | BAA5 | BAA4 | BAA3 | BAA2 | BAA1 | BAA0 | |
| BAMRA | BAMA31 | BAMA30 | BAMA29 | BAMA28 | BAMA27 | BAMA26 | BAMA25 | BAMA24 | |
| | BAMA23 | BAMA22 | BAMA21 | BAMA20 | BAMA19 | BAMA18 | BAMA17 | BAMA16 | |
| | BAMA15 | BAMA14 | BAMA13 | BAMA12 | BAMA11 | BAMA10 | BAMA9 | BAMA8 | |
| | BAMA7 | BAMA6 | BAMA5 | BAMA4 | BAMA3 | BAMA2 | BAMA1 | BAMA0 | |
| BBRA | — | — | — | — | — | CPA[2:0] | | | |
| | CDA[1:0] | | IDA[1:0] | | RWA[1:0] | | SZA[1:0] | | |
| BDRA | BDA31 | BDA30 | BDA29 | BDA28 | BDA27 | BDA26 | BDA25 | BDA24 | |
| | BDA23 | BDA22 | BDA21 | BDA20 | BDA19 | BDA18 | BDA17 | BDA16 | |
| | BDA15 | BDA14 | BDA13 | BDA12 | BDA11 | BDA10 | BDA9 | BDA8 | |
| | BDA7 | BDA6 | BDA5 | BDA4 | BDA3 | BDA2 | BDA1 | BDA0 | |
| BDMRA | BDMA31 | BDMA30 | BDMA29 | BDMA28 | BDMA27 | BDMA26 | BDMA25 | BDMA24 | |
| | BDMA23 | BDMA22 | BDMA21 | BDMA20 | BDMA19 | BDMA18 | BDMA17 | BDMA16 | |
| | BDMA15 | BDMA14 | BDMA13 | BDMA12 | BDMA11 | BDMA10 | BDMA9 | BDMA8 | |
| | BDMA7 | BDMA6 | BDMA5 | BDMA4 | BDMA3 | BDMA2 | BDMA1 | BDMA0 | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| BARB | BAB31 | BAB30 | BAB29 | BAB28 | BAB27 | BAB26 | BAB25 | BAB24 | UBC |
| | BAB23 | BAB22 | BAB21 | BAB20 | BAB19 | BAB18 | BAB17 | BAB16 | |
| | BAB15 | BAB14 | BAB13 | BAB12 | BAB11 | BAB10 | BAB9 | BAB8 | |
| | BAB7 | BAB6 | BAB5 | BAB4 | BAB3 | BAB2 | BAB1 | BAB0 | |
| BAMRB | BAMB31 | BAMB30 | BAMB29 | BAMB28 | BAMB27 | BAMB26 | BAMB25 | BAMB24 | |
| | BAMB23 | BAMB22 | BAMB21 | BAMB20 | BAMB19 | BAMB18 | BAMB17 | BAMB16 | |
| | BAMB15 | BAMB14 | BAMB13 | BAMB12 | BAMB11 | BAMB10 | BAMB9 | BAMB8 | |
| | BAMB7 | BAMB6 | BAMB5 | BAMB4 | BAMB3 | BAMB2 | BAMB1 | BAMB0 | |
| BBRB | — | — | — | — | — | CPB[2:0] | | | |
| | CDB[1:0] | | IDB[1:0] | | RWB[1:0] | | SZB[1:0] | | |
| BDRB | BDB31 | BDB30 | BDB29 | BDB28 | BDB27 | BDB26 | BDB25 | BDB24 | |
| | BDB23 | BDB22 | BDB21 | BDB20 | BDB19 | BDB18 | BDB17 | BDB16 | |
| | BDB15 | BDB14 | BDB13 | BDB12 | BDB11 | BDB10 | BDB9 | BDB8 | |
| | BDB7 | BDB6 | BDB5 | BDB4 | BDB3 | BDB2 | BDB1 | BDB0 | |
| BDMRB | BDMB31 | BDMB30 | BDMB29 | BDMB28 | BDMB27 | BDMB26 | BDMB25 | BDMB24 | |
| | BDMB23 | BDMB22 | BDMB21 | BDMB20 | BDMB19 | BDMB18 | BDMB17 | BDMB16 | |
| | BDMB15 | BDMB14 | BDMB13 | BDMB12 | BDMB11 | BDMB10 | BDMB9 | BDMB8 | |
| | BDMB7 | BDMB6 | BDMB5 | BDMB4 | BDMB3 | BDMB2 | BDMB1 | BDMB0 | |
| BRCR | — | — | — | — | — | — | — | — | |
| | — | — | UTRGW[1:0] | | UBIDB | — | UBIDA | — | |
| | SCMFCA | SCMFCB | SCMFDA | SCMFDB | PCTE | PCBA | — | — | |
| | DBEA | PCBB | DBEB | — | SEQ | — | — | ETBE | |
| BRSR | SVF | — | — | — | BSA27 | BSA26 | BSA25 | BSA24 | |
| | BSA23 | BSA22 | BSA21 | BSA20 | BSA19 | BSA18 | BSA17 | BSA16 | |
| | BSA15 | BSA14 | BSA13 | BSA12 | BSA11 | BSA10 | BSA9 | BSA8 | |
| | BSA7 | BSA6 | BSA5 | BSA4 | BSA3 | BSA2 | BSA1 | BSA0 | |
| BRDR | DVF | — | — | — | BDA27 | BDA26 | BDA25 | BDA24 | |
| | BDA23 | BDA22 | BDA21 | BDA20 | BDA19 | BDA18 | BDA17 | BDA16 | |
| | BDA15 | BDA14 | BDA13 | BDA12 | BDA11 | BDA10 | BDA9 | BDA8 | |
| | BDA7 | BDA6 | BDA5 | BDA4 | BDA3 | BDA2 | BDA1 | BDA0 | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| BETR | — | — | — | — | BET[11:8] | | | | UBC |
| | BET[7:0] | | | | | | | | |

## 25.3 Register States in Each Operating Mode

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| SCSMR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | SCI (Channel 0) |
| SCBRR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSCR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCTDR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSSR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCRDR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSDCR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSPTR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSMR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | SCI (Channel 1) |
| SCBRR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSCR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCTDR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSSR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCRDR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSDCR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSPTR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSMR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | SCI (Channel 2) |
| SCBRR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSCR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCTDR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSSR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCRDR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSDCR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SCSPTR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCR_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | MTU2 |
| TCR_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TMDR_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TMDR_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORH_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| TIORL_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | MTU2 |
| TIORH_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORL_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TOER | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGCR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TOCR1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TOCR2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNT_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNT_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCDR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TDDR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRA_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRB_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRA_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRB_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCBR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRC_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRD_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRC_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRD_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSR_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSR_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TITCR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TITCNT | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TBTER | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TDER | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TOLBR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TBTM_3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| TBTM_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | MTU2 |
| TADCR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TADCORA_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TADCORB_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TADCOBRA_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TADCOBRB_4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TWCR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSTR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSYR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCSYSTR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TRWER | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TMDR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORH_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORL_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNT_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRA_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRB_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRC_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRD_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRE_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRF_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER2_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSR2_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TBTM_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TMDR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIOR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| TSR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | MTU2 |
| TCNT_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRA_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRB_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TICCR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TMDR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIOR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSR_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNT_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRA_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRB_2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTU_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRU_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCRU_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORU_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTV_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRV_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCRV_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORV_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTW_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRW_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCRW_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORW_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSR_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER_5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSTR5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTCMPCLR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCR_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | MTU2S |
| TCR_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| TMDR_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | MTU2S |
| TMDR_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORH_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORL_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORH_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORL_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TOERS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGCRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TOCR1S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TOCR2S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNT_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNT_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCDRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TDDRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRA_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRB_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRA_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRB_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTSS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCBRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRC_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRD_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRC_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRD_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSR_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSR_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TITCRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TITCNTS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TBTERS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| TDERS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | MTU2S |
| TOLBRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TBTM_3S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TBTM_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TADCRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TADCORA_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TADCORB_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TADCOBRA_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TADCOBRB_4S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSYCRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TWCRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSTRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSYRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TRWERS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTU_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRU_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCRU_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORU_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTV_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRV_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCRV_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORV_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTW_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TGRW_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCRW_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIORW_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSR_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TIER_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TSTR_5S | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TCNTCMPCLRS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| FCCS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | FLASH |
| FPCS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| FECS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| FKEY | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| FMATS | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| FTDAR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| DTCERA | Initialized | Retained | Retained | Initialized | Retained | Retained | DTC |
| DTCERB | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| DTCERC | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| DTCERD | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| DTCERE | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| DTCCR | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| DTCVBR | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| ICCR1 | Initialized | Retained | Retained | Initialized | Retained | Retained | $I^2C2$ |
| ICCR2 | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| ICMR | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| ICIER | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| ICSR | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| SAR | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| ICDRT | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| ICDRR | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| NF2CYC | Initialized | Retained | Retained | Initialized | Retained | Retained | |
| SSCRH | Initialized | Retained | Initialized | Initialized | Initialized | Retained | SSU |
| SSCRL | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSMR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSER | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSSR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSCR2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSTDR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSTDR1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSTDR2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| SSTDR3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | SSU |
| SSRDR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSRDR1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSRDR2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| SSRDR3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| CMSTR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | CMT |
| CMCSR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| CMCNT_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| CMCOR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| CMCSR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| CMCNT_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| CMCOR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ICSR1 | Initialized | Retained | Retained | Initialized | — | Retained | POE |
| OCSR1 | Initialized | Retained | Retained | Initialized | — | Retained | |
| ICSR2 | Initialized | Retained | Retained | Initialized | — | Retained | |
| OCSR2 | Initialized | Retained | Retained | Initialized | — | Retained | |
| ICSR3 | Initialized | Retained | Retained | Initialized | — | Retained | |
| SPOER | Initialized | Retained | Retained | Initialized | — | Retained | |
| POECR1 | Initialized | Retained | Retained | Initialized | — | Retained | |
| POECR2 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PADRL | Initialized | Retained | Retained | Initialized | — | Retained | I/O |
| PAIORL | Initialized | Retained | Retained | Initialized | — | Retained | PFC |
| PACRL4 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PACRL3 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PACRL2 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PACRL1 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PAPRL | Initialized | Retained | Retained | Initialized | — | Retained | I/O |
| PBDRL | Initialized | Retained | Retained | Initialized | — | Retained | |
| PBIORL | Initialized | Retained | Retained | Initialized | — | Retained | PFC |
| PBCRL2 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PBCRL1 | Initialized | Retained | Retained | Initialized | — | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| PBPRL | Initialized | Retained | Retained | Initialized | — | Retained | I/O |
| PDDRL | Initialized | Retained | Retained | Initialized | — | Retained | |
| PDIORL | Initialized | Retained | Retained | Initialized | — | Retained | PFC |
| PDCRL3 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PDCRL2 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PDCRL1 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PDPRL | Initialized | Retained | Retained | Initialized | — | Retained | I/O |
| PEDRH | Initialized | Retained | Retained | Initialized | — | Retained | |
| PEDRL | Initialized | Retained | Retained | Initialized | — | Retained | |
| PEIORH | Initialized | Retained | Retained | Initialized | — | Retained | PFC |
| PEIORL | Initialized | Retained | Retained | Initialized | — | Retained | |
| PECRH2 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PECRH1 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PECRL4 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PECRL3 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PECRL2 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PECRL1 | Initialized | Retained | Retained | Initialized | — | Retained | |
| PEPRH | Initialized | Retained | Retained | Initialized | — | Retained | I/O |
| PEPRL | Initialized | Retained | Retained | Initialized | — | Retained | |
| IFCR | Initialized | Retained | Retained | Initialized | — | Retained | PFC |
| PFDRL | Initialized | Retained | Retained | Initialized | — | Retained | I/O |
| ADCR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | A/D |
| ADSR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | (Channel 0) |
| ADSTRGR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADANSR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR2 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR3 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR4 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR5 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| ADDR6 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | A/D (Channel 0) |
| ADDR7 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADCR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | A/D (Channel 1) |
| ADSR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADSTRGR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADANSR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR8 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR9 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR10 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR11 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR12 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR13 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR14 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ADDR15 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| MCR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | RCAN-ET |
| GSR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| BCR1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| BCR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| IRR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| IMR | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TEC/REC | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TXPR1, TXPR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TXCR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| TXACK0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| ABACK0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| RXPR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| RFPR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| MBIMR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| UMSR0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| MB[0]. CONTROL0H | — | Retained | — | — | — | Retained | |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| MB[0]. CONTROL0L | — | Retained | — | — | — | Retained | RCAN-ET |
| MB[0]. LAFMH | — | Retained | — | — | — | Retained | |
| MB[0]. LAFML | — | Retained | — | — | — | Retained | |
| MB[0]. MSG_DATA[0] | — | Retained | — | — | — | Retained | |
| MB[0]. MSG_DATA[1] | — | Retained | — | — | — | Retained | |
| MB[0]. MSG_DATA[2] | — | Retained | — | — | — | Retained | |
| MB[0]. MSG_DATA[3] | — | Retained | — | — | — | Retained | |
| MB[0]. MSG_DATA[4] | — | Retained | — | — | — | Retained | |
| MB[0]. MSG_DATA[5] | — | Retained | — | — | — | Retained | |
| MB[0]. MSG_DATA[6] | — | Retained | — | — | — | Retained | |
| MB[0]. MSG_DATA[7] | — | Retained | — | — | — | Retained | |
| MB[0]. CONTROL1H | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| MB[0]. CONTROL1L | Initialized | Retained | Initialized | Initialized | Initialized | Retained | |
| MB[1] | Same as MB[0] | | | | | | |
| MB[2] | Same as MB[0] | | | | | | |
| MB[3] | Same as MB[0] | | | | | | |
| ↓ | (Repeat) | | | | | | |
| MB[13] | Same as MB[0] | | | | | | |
| MB[14] | Same as MB[0] | | | | | | |
| MB[15] | Same as MB[0] | | | | | | |
| FRQCR | Initialized*1 | Retained | Retained | Initialized | — | Retained | CPG |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| STBCR1 | Initialized | Retained | Retained | Initialized | — | Retained | Power-down modes |
| STBCR2 | Initialized | Retained | Retained | Initialized | — | Retained | |
| STBCR3 | Initialized | Retained | Retained | Initialized | — | Retained | |
| STBCR4 | Initialized | Retained | Retained | Initialized | — | Retained | |
| STBCR5 | Initialized | Retained | Retained | Initialized | — | Retained | |
| STBCR6 | Initialized | Retained | Retained | Initialized | — | Retained | |
| WTCNT | Initialized*1 | Retained | Retained | Initialized | — | Retained | WDT |
| WTCSR | Initialized*1 | Retained | Retained | Initialized | — | Retained | |
| OSCCR | Initialized*2 | Retained | Retained*3 | Initialized | — | Retained | CPG |
| RAMCR | Initialized | Retained | Retained | Initialized | — | Retained | Power-down modes |
| BSCEHR | Initialized | Retained | Retained | Initialized | — | Retained | BSC |
| ICR0 | Initialized | Initialized | Retained | Initialized | — | Retained | INTC |
| IRQCR | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IRQSR | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRA | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRD | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRE | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRF | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRH | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRI | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRJ | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRK | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRL | Initialized | Initialized | Retained | Initialized | — | Retained | |
| IPRM | Initialized | Initialized | Retained | Initialized | — | Retained | |
| CMNCR | Initialized | Retained | Retained | Initialized | — | Retained | BSC |
| CS0BCR | Initialized | Retained | Retained | Initialized | — | Retained | |
| CS1BCR | Initialized | Retained | Retained | Initialized | — | Retained | |
| CS0WCR | Initialized | Retained | Retained | Initialized | — | Retained | |
| CS1WCR | Initialized | Retained | Retained | Initialized | — | Retained | |
| RAMER | Initialized | Initialized | Retained | Initialized | Retained | Retained | FLASH |

| Register Abbreviation | Power-on reset | Manual reset | Software Standby | Deep Software Standby | Module Standby | Sleep | Module |
|---|---|---|---|---|---|---|---|
| BARA | Initialized | Retained | Retained | Initialized | Initialized | Retained | UBC |
| BAMRA | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BBRA | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BDRA | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BDMRA | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BARB | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BAMRB | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BBRB | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BDRB | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BDMRB | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BRCR | Initialized | Retained | Retained | Initialized | Initialized | Retained | |
| BRSR | Initialized | Initialized | Retained | Initialized | Initialized | Retained | |
| BRDR | Initialized | Initialized | Retained | Initialized | Initialized | Retained | |
| BETR | Initialized | Retained | Retained | Initialized | Initialized | Retained | |

Notes: 1. Not initialized by a WDT power-on reset.
2. The OSCSTOP bit is not initialized by a WDT power-on reset.
3. The OSCSTOP bit is initialized.

# Section 26 Electrical Characteristics

Note: The values in this hardware manual are provisional. They are subject to change without notice.

## 26.1 Absolute Maximum Ratings

Table 26.1 lists the absolute maximum ratings.

**Table 26.1 Absolute Maximum Ratings**

| Item | | Symbol | Value | Unit |
|---|---|---|---|---|
| Power supply voltage | | $V_{CC}$ | –0.3 to +7.0 | V |
| Input voltage (except analog input pins) | | $V_{in}$ | –0.3 to $V_{CC}$ +0.3 | V |
| Analog power supply voltage | | $AV_{CC}$ | –0.3 to +7.0 | V |
| Analog reference voltage | | $AV_{refh}$ | –0.3 to $AV_{CC}$ +0.3 | V |
| Analog input voltage | | $V_{an}$ | –0.3 to $AV_{CC}$ +0.3 | V |
| Operating temperature | Consumer applications | $T_{opr}$ | –20 to +85 | °C |
| | Industrial applications | | –40 to +85 | °C |
| Storage temperature | | $T_{stg}$ | –55 to +125 | °C |

[Operating Precaution]

Operating the LSI in excess of the absolute maximum ratings may result in permanent damage.

## 26.2 DC Characteristics

Tables 26.2 and 26.3 list DC characteristics.

### Table 26.2 DC Characteristics

Conditions: $V_{CC}$ = 3.0 V to 3.6 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$,
$V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V, $T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Input high-level voltage (except Schmitt trigger input voltage) | $\overline{RES}$, $\overline{MRES}$, NMI, FWE, MD1, MD0, $\overline{ASEMD0}$, EXTAL | $V_{IH}$ | $V_{CC}$–0.5 | — | $V_{CC}$+0.3 | V | |
| | Analog ports | | 2.2 | — | $AV_{CC}$+0.3 | V | |
| | Other input pins | | 2.2 | — | $V_{CC}$+0.3 | V | |
| Input low-level voltage (except Schmitt trigger input voltage) | $\overline{RES}$, $\overline{MRES}$, NMI, FWE, MD1, MD0, $\overline{ASEMD0}$, EXTAL | $V_{IL}$ | –0.3 | — | 0.5 | V | |
| | Other input pins | | –0.3 | — | 0.8 | V | |
| Schmitt trigger input voltage | IRQ3 to IRQ0, $\overline{POE8}$, $\overline{POE6}$ to $\overline{POE4}$, $\overline{POE2}$ to $\overline{POE0}$ TCLKA to TCLKD, TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D, TIOC4A to TIOC4D, TIC5U, TIC5V, TIC5W, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS, TIC5US, TIC5VS, TIC5WS, SCK0 to SCK2, RXD0 to RXD2, SSCK, $\overline{SCS}$, SSI, SSO, SCL, SDA | $V_{T+}$ | $V_{CC}$–0.5 | — | — | V | |
| | | $V_{T-}$ | — | — | 0.5 | V | |
| | | $V_{T+}$–$V_{T-}$ | 0.2 | — | — | V | |
| Input leak current | All input pins (except $\overline{ASEMD0}$) | $\lvert I_{in} \rvert$ | — | — | 1.0 | μA | |

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Input pull-up MOS current | $\overline{\text{ASEMD0}}$ | $-I_{pu}$ | — | — | 350 | μA | $V_{in}$ = 0 V |
| Three-state leak current (OFF state) | Ports A, B, D, E | $\lvert I_{tsi} \rvert$ | — | — | 1.0 | μA | |
| Output high-level voltage | All output pins | $V_{OH}$ | $V_{CC}$–0.5 | — | — | V | $I_{OH}$ = –200 μA |
| | | | $V_{CC}$–1.0 | — | — | V | $I_{OH}$ = –1 mA |
| | TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS | | $V_{CC}$–1.0 | — | — | V | $I_{OH}$ = –5 mA |
| | PE9, PE11 to PE21 | | $V_{CC}$–2.0 | — | — | V | $I_{OH}$ = –5 mA |
| Output low-level voltage | All output pins | $V_{OL}$ | — | — | 0.4 | V | $I_{OL}$ = 1.6 mA |
| | SCL, SDA | | — | — | 0.4 | V | $I_{OL}$ = 3 mA |
| | | | — | — | 0.5 | V | $I_{OL}$ = 8 mA |
| | TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS | | — | — | 0.9 | V | $I_{OL}$ = 15 mA |
| | PE9, PE11 to PE21 | | — | — | 2.0 | V | $I_{OL}$ = 15 mA |
| Input capacitance | All input pins | $C_{in}$ | — | — | 20 | pF | $V_{in}$ = 0 V<br>f = 1 MHz<br>Ta = 25°C |
| Supply current | Normal operation | $I_{CC}$ | — | 80 | 105 | mA | Iϕ = 80 MHz<br>Bϕ = 40 MHz<br>Pϕ = 40 MHz<br>MPϕ = 40 MHz<br>MIϕ = 80 MHz |
| | Sleep | | — | 55 | 85 | mA | Bϕ = 40 MHz<br>Pϕ = 40 MHz<br>MPϕ = 40 MHz<br>MIϕ = 80 MHz |

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Supply current | Software standby | $I_{CC}$ | — | 8 | 20 | mA | $T_a \leq 50°C$ |
| | | | — | — | 30 | mA | $50°C < T_a$ |
| | Deep software standby | | — | 2 | 10 | µA | $T_a \leq 50°C$ |
| | | | — | — | 40 | µA | $50°C < T_a$ |
| Analog power supply current | During A/D conversion | $AI_{CC}$ | — | 3 | 5 | mA | The value per A/D converter module. |
| | Waiting for A/D conversion | | — | — | 0.1 | mA | |
| | Standby | | — | — | 15 | µA | |
| Reference power supply current | During A/D conversion | $AI_{refh}$ | — | — | 2 | mA | The value per A/D converter module. |
| | Waiting for A/D conversion | | — | — | 2 | mA | |
| | Standby | | — | — | 2.5 | µA | |
| RAM standby voltage | | VRAM | 2 | — | — | V | $V_{CC}$ |

[Operating Precautions]

1. When the A/D converter is not used, do not leave the $AV_{CC}$, $AV_{SS}$, $AV_{refh}$, and $AV_{refl}$ pins open.
2. The supply current was measured when $V_{IH}$ (Min.) = $V_{CC}$ – 0.5 V, $V_{IL}$ (Max.) = 0.5 V, with all output pins unloaded.

## Table 26.3 DC Characteristics

Conditions: $V_{CC}$ = 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$,
$V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V, $T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Input high-level voltage (except Schmitt trigger input voltage) | $\overline{RES}$, $\overline{MRES}$, NMI, FWE, MD1, MD0, $\overline{ASEMD0}$, EXTAL | $V_{IH}$ | $V_{CC}$–0.6 | — | $V_{CC}$+0.3 | V | |
| | Analog ports | | 2.2 | — | $AV_{CC}$+0.3 | V | |
| | Other input pins | | 2.2 | — | $V_{CC}$+0.3 | V | |
| Input low-level voltage (except Schmitt trigger input voltage) | $\overline{RES}$, $\overline{MRES}$, NMI, FWE, MD1, MD0, $\overline{ASEMD0}$, EXTAL | $V_{IL}$ | –0.3 | — | 0.4 | V | |
| | Other input pins | | –0.3 | — | 0.8 | V | |
| Schmitt trigger input voltage | IRQ3 to IRQ0, $\overline{POE8}$, $\overline{POE6}$ to $\overline{POE4}$, $\overline{POE2}$ to $\overline{POE0}$ TCLKA to TCLKD, TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D, TIOC4A to TIOC4D, TIC5U, TIC5V, TIC5W, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS, TIC5US, TIC5VS, TIC5WS, SCK0 to SCK2, RXD0 to RXD2, SSCK, $\overline{SCS}$, SSI, SSO, SCL, SDA | $V_{T+}$ | $V_{CC}$–0.5 | — | — | V | |
| | | $V_{T-}$ | — | — | 1.0 | V | |
| | | $V_{T+}$–$V_{T-}$ | 0.4 | — | — | V | |
| Input leak current | All input pins (except $\overline{ASEMD0}$) | $\lvert I_{in} \rvert$ | — | — | 1.0 | μA | |
| Input pull-up MOS current | $\overline{ASEMD0}$ | $-I_{pu}$ | — | — | 800 | μA | $V_{in}$ = 0 V |

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Three-state leak current (OFF state) | Ports A, B, D, E | $\lvert I_{tsi} \rvert$ | — | — | 1.0 | μA | |
| Output high-level voltage | All output pins | $V_{OH}$ | $V_{CC}$–0.5 | — | — | V | $I_{OH}$ = –200 μA |
| | | | $V_{CC}$–1.0 | — | — | V | $I_{OH}$ = –1 mA |
| | TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS | | $V_{CC}$–1.0 | — | — | V | $I_{OH}$ = –5 mA |
| | PE9, PE11 to PE21 | | $V_{CC}$–2.0 | — | — | V | $I_{OH}$ = –5 mA |
| Output low-level voltage | All output pins | $V_{OL}$ | — | — | 0.4 | V | $I_{OL}$ = 1.6 mA |
| | SCL, SDA | | — | — | 0.4 | V | $I_{OL}$ = 3 mA |
| | | | — | — | 0.5 | V | $I_{OL}$ = 8 mA |
| | TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS | | — | — | 1.4 | V | $I_{OL}$ = 15 mA |
| | PE9, PE11 to PE21 | | — | — | 1.5 | V | $I_{OL}$ = 15 mA |
| Input capacitance | All input pins | $C_{in}$ | — | — | 20 | pF | $V_{in}$ = 0 V<br>f = 1 MHz<br>Ta = 25°C |
| Supply current | Normal operation | $I_{CC}$ | — | 80 | 105 | mA | Iϕ = 80 MHz<br>Bϕ = 40 MHz<br>Pϕ = 40 MHz<br>MPϕ = 40 MHz<br>MIϕ = 80 MHz |
| | Sleep | | — | 55 | 85 | mA | Bϕ = 40 MHz<br>Pϕ = 40 MHz<br>MPϕ = 40 MHz<br>MIϕ = 80 MHz |
| | Software standby | | — | 8 | 20 | mA | $T_a \leq 50°C$ |
| | | | — | — | 30 | mA | $50°C < T_a$ |

| Item | | Symbol | Min. | Typ. | Max. | Unit | Measurement Conditions |
|---|---|---|---|---|---|---|---|
| Supply current | Deep software standby | $I_{CC}$ | — | 2 | 10 | µA | $T_a \leq 50°C$ |
| | | | — | — | 40 | µA | $50°C < T_a$ |
| Analog power supply current | During A/D conversion | $AI_{CC}$ | — | 3 | 5 | mA | The value per A/D converter module. |
| | Waiting for A/D conversion | | — | — | 0.1 | mA | |
| | Standby | | — | — | 15 | µA | |
| Reference power supply current | During A/D conversion | $AI_{refh}$ | — | — | 2 | mA | The value per A/D converter module. |
| | Waiting for A/D conversion | | — | — | 2 | mA | |
| | Standby | | — | — | 2.5 | µA | |
| RAM standby voltage | | VRAM | 2 | — | — | V | $V_{CC}$ |

[Operating Precautions]

1. When the A/D converter is not used, do not leave the $AV_{CC}$, $AV_{SS}$, $AV_{refh}$, and $AV_{refl}$ pins open.
2. The supply current was measured when $V_{IH}$ (Min.) = $V_{CC}$ – 0.5 V, $V_{IL}$ (Max.) = 0.5 V, with all output pins unloaded.

## Table 26.4 Permissible Output Current Values

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V,
$AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Output low-level permissible current (per pin) | $I_{OL}$ | — | — | 2.0* | mA |
| Output low-level permissible current (total) | $\Sigma I_{OL}$ | — | — | 110 | mA |
| Output high-level permissible current (per pin) | $-I_{OH}$ | — | — | 2.0* | mA |
| Output high-level permissible current (total) | $\Sigma -I_{OH}$ | — | — | 35 | mA |

[Operating Precaution]

To assure LSI reliability, do not exceed the output values listed in table 26.4.

Note: * $I_{OL}$ = 15 mA (Max.)/$-I_{OH}$ = 5 mA (Max.) for pins PE9, and PE11 to PE21. $I_{OL}$ = 8 mA (Max.) about pins SCL and SDA. However, at most six pins are permitted to have simultaneously $I_{OL}/-I_{OH} > 2.0$ mA among these pins.

## 26.3 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 26.5 Maximum Operating Frequency**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V,
$AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|---|
| Operating frequency | CPU (Iϕ) | f | 10 | — | 80 | MHz | |
| | External bus (Bϕ) | | 10 | — | 40 | | |
| | Peripheral module (Pϕ) | | 10 | — | 40 | | |
| | MTU2 (MPϕ) | | 10 | — | 40 | | |
| | MTU2S (MIϕ) | | 10 | — | 80 | | |

### 26.3.1 Clock Timing

**Table 26.6 Clock Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V,
$AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| EXTAL clock input frequency | $f_{EX}$ | 5 | 12.5 | MHz | Figure 26.1 |
| EXTAL clock input cycle time | $t_{EXcyc}$ | 80 | 200 | ns | |
| EXTAL clock input low pulse width | $t_{EXL}$ | 20 | — | ns | |
| EXTAL clock input high pulse width | $t_{EXH}$ | 20 | — | ns | |
| EXTAL clock input rise time | $t_{EXr}$ | — | 5 | ns | |
| EXTAL clock input fall time | $t_{EXf}$ | — | 5 | ns | |
| CK clock output frequency | $f_{OP}$ | 10 | 40 | MHz | Figure 26.2 |
| CK clock output cycle time | $t_{cyc}$ | 25 | 100 | ns | |
| CK clock output low pulse width | $t_{CKL}$ | $1/2t_{cyc}$–7.5 | — | ns | |
| CK clock output high pulse width | $t_{CKH}$ | $1/2t_{cyc}$–7.5 | — | ns | |
| CK clock output rise time | $t_{CKr}$ | — | 5 | ns | |
| CK clock output fall time | $t_{CKf}$ | — | 5 | ns | |
| Power-on oscillation settling time | $t_{OSC1}$ | 10 | — | ms | Figure 26.3 |
| Standby return oscillation settling time 1 | $t_{OSC2}$ | 10 | — | ms | Figure 26.4 |
| Standby return oscillation settling time 2 | $t_{OSC3}$ | 10 | — | ms | Figure 26.5 |

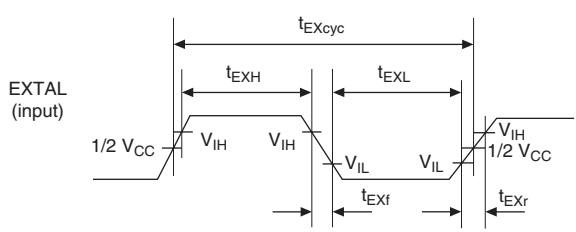



**Figure 26.1 EXTAL Clock Input Timing**

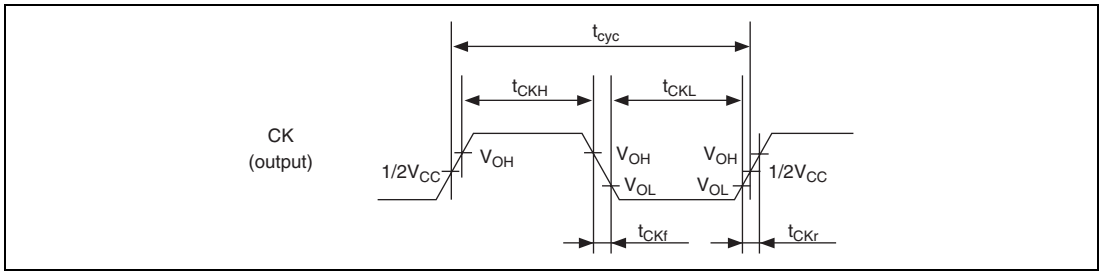



**Figure 26.2 CK Clock Output Timing**

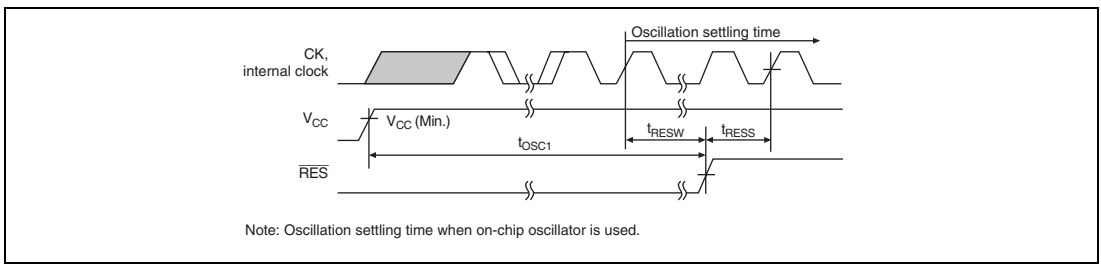



**Figure 26.3 Power-On Oscillation Settling Timing**

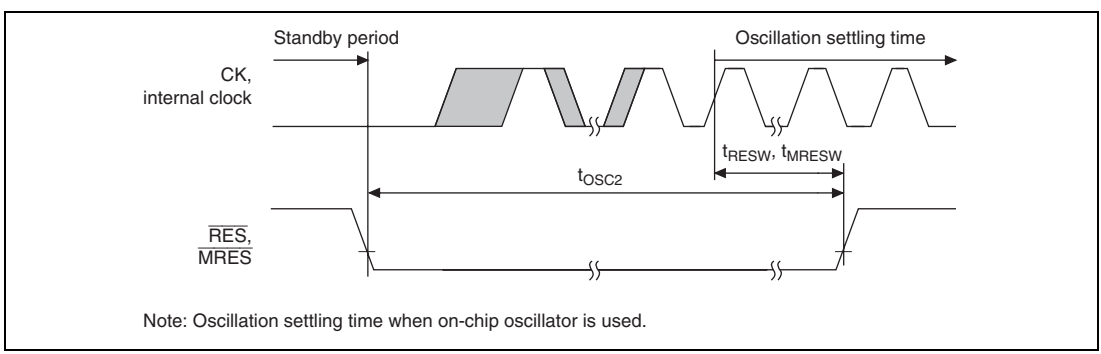
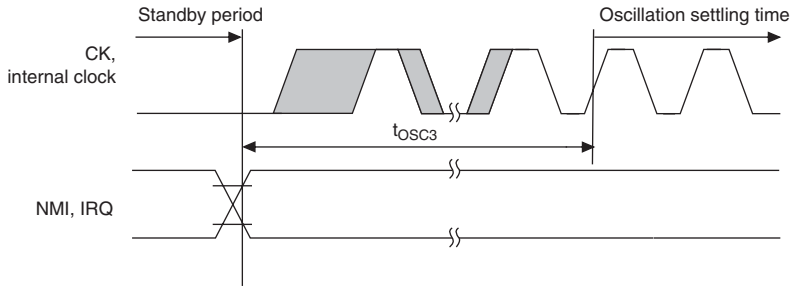



**Figure 26.4 Oscillation Settling Timing on Return from Standby (Return by Reset)**




Note: Oscillation settling time when on-chip oscillator is used.

**Figure 26.5 Oscillation Settling Timing on Return from Standby (Return by NMI or IRQ)**

### 26.3.2 Control Signal Timing

**Table 26.7 Control Signal Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V,
$AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| $\overline{RES}$ pulse width | $t_{RESW}$ | 20*2 | — | $t_{Bcyc}$*4 | Figures 26.3, 26.4, |
| $\overline{RES}$ setup time*1 | $t_{RESS}$ | 65 | — | ns | 26.6, 26.7 |
| $\overline{RES}$ hold time | $t_{RESH}$ | 15 | — | ns | |
| $\overline{MRES}$ pulse width | $t_{MRESW}$ | 20*3 | — | $t_{Bcyc}$*4 | |
| $\overline{MRES}$ setup time*1 | $t_{MRESS}$ | 25 | — | ns | |
| $\overline{MRES}$ hold time | $t_{MRESH}$ | 15 | — | ns | |
| MD1, MD0, FWE setup time | $t_{MDS}$ | 20 | — | $t_{Bcyc}$*4 | Figure 26.6 |
| $\overline{BREQ}$ setup time | $t_{BREQS}$ | $1/2t_{Bcyc}$ + 15 | — | ns | Figure 26.9 |
| $\overline{BREQ}$ hold time | $t_{BREQH}$ | $1/2t_{Bcyc}$ + 10 | — | ns | |
| NMI setup time*1 | $t_{NMIS}$ | 60 | — | ns | Figure 26.7 |
| NMI hold time | $t_{NMIH}$ | 10 | — | ns | |
| IRQ3 to IRQ0 setup time*1 | $t_{IRQS}$ | 35 | — | ns | |
| IRQ3 to IRQ0 hold time | $t_{IRQH}$ | 35 | — | ns | |
| $\overline{IRQOUT}$ output delay time | $t_{IRQOD}$ | — | 100 | ns | Figure 26.8 |
| $\overline{BACK}$ delay time | $t_{BACKD}$ | — | $1/2t_{Bcyc}$ + 20 | ns | Figures 26.9, 26.10 |
| Bus tri-state delay time | $t_{BOFF}$ | 0 | 100 | ns | |
| Bus buffer on time | $t_{BON}$ | 0 | 100 | ns | |

Notes:
1. The $\overline{RES}$, $\overline{MRES}$, NMI, $\overline{BREQ}$, and IRQ3 to IRQ0 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 is delayed until the next rising edge of the clock.
2. In standby mode, $t_{RESW}$ = $t_{OSC2}$ (10 ms).
3. In standby mode, $t_{MRESW}$ = $t_{OSC2}$ (10 ms).
4. $t_{Bcyc}$ indicates external bus clock cycle time (Bϕ = CK).

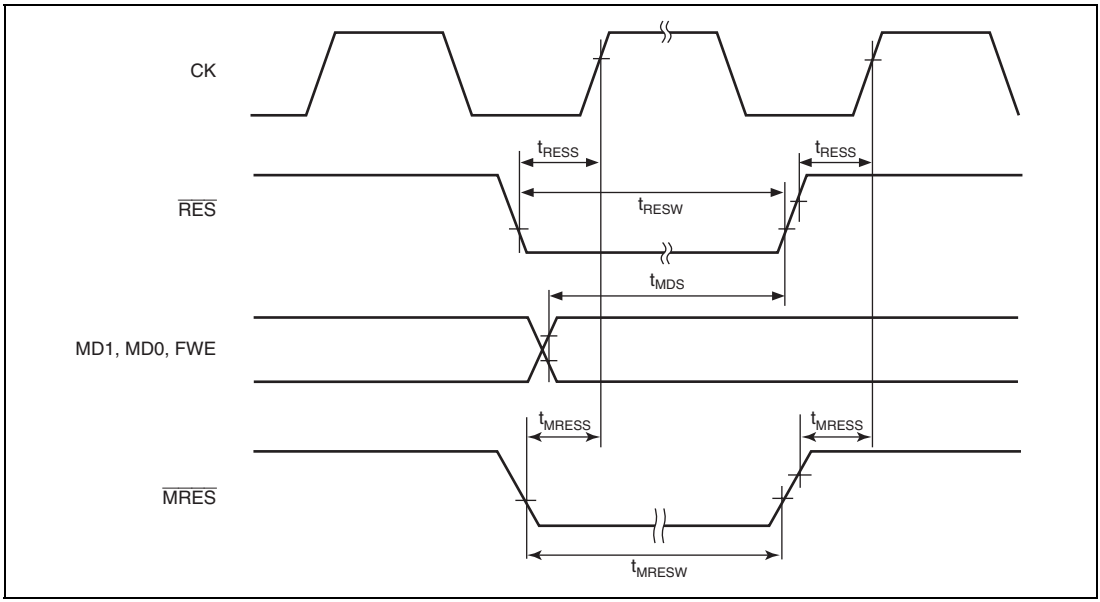



**Figure 26.6 Reset Input Timing**

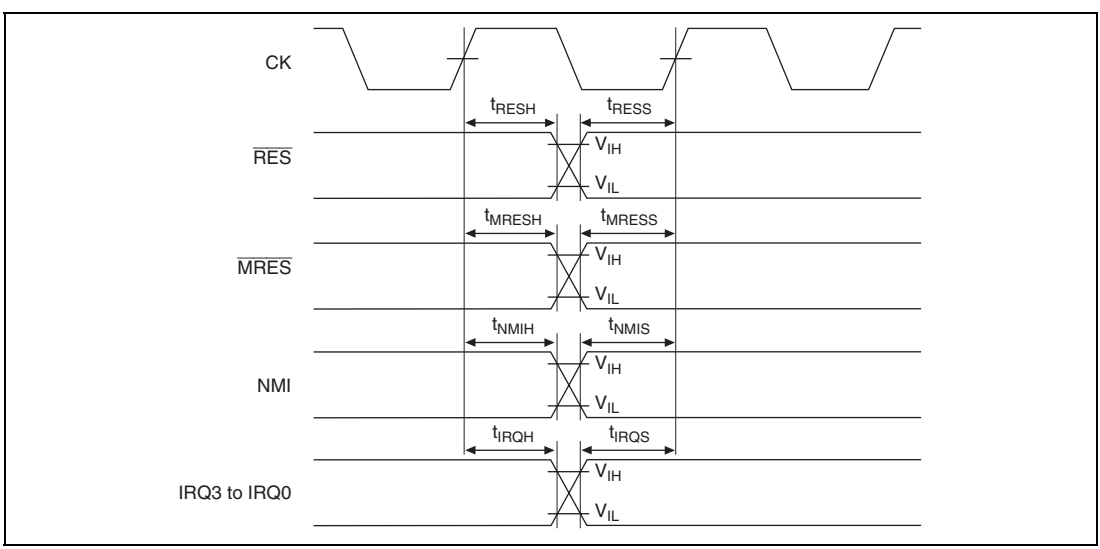



**Figure 26.7 Interrupt Signal Input Timing**

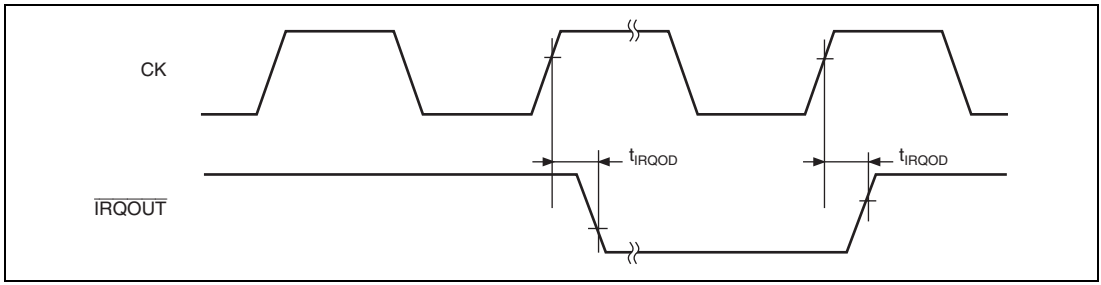



**Figure 26.8 Interrupt Signal Output Timing**

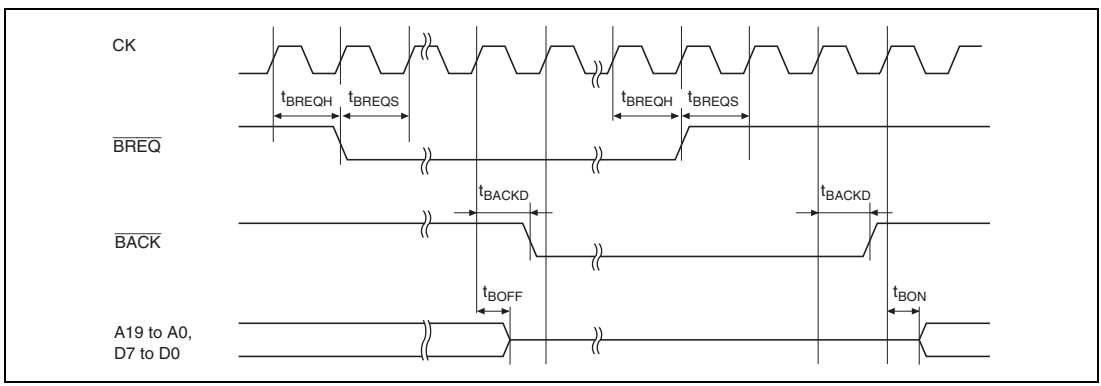



**Figure 26.9 Bus Release Timing**

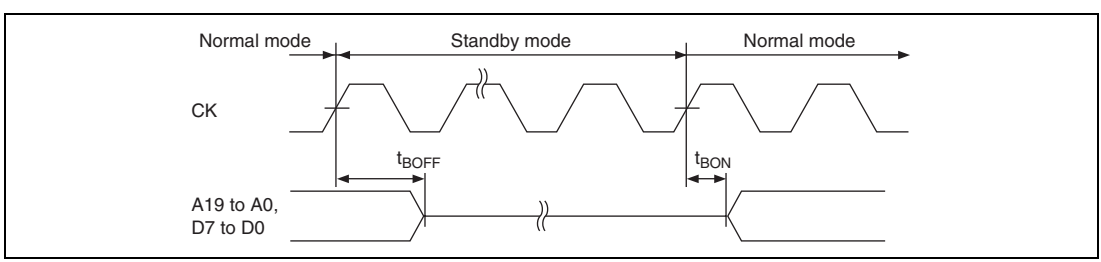



**Figure 26.10 Pin Driving Timing in Standby Mode**

### 26.3.3 AC Bus Timing

**Table 26.8 Bus Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V,
$AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| Address delay time 1 | $t_{AD1}$ | 1 | 20 | ns | Figures 26.11 to 26.15 |
| Address setup time | $t_{AS}$ | 0 | — | ns | Figures 26.11 to 26.14 |
| Address hold time | $t_{AH}$ | 0 | — | ns | Figures 26.11 to 26.14 |
| $\overline{CS}$ delay time | $t_{CSD}$ | 1 | 18 | ns | Figures 26.11 to 26.15 |
| $\overline{CS}$ setup time | $t_{CSS}$ | 0 | — | ns | Figures 26.11 to 26.14 |
| $\overline{CS}$ hold time | $t_{CSH}$ | 0 | — | ns | Figures 26.11 to 26.14 |
| Read strobe delay time | $t_{RSD}$ | $1/2t_{Bcyc}$ + 1 | $1/2t_{Bcyc}$ + 18 | ns | Figures 26.11 to 26.15 |
| Read data setup time 1 | $t_{RDS1}$ | $1/2t_{Bcyc}$ + 18 | — | ns | Figures 26.11 to 26.15 |

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| Read data hold time 1 | $t_{RDH1}$ | 0 | — | ns | Figures 26.11 to 26.15 |
| Read data access time | $t_{ACC}$ *2 | $t_{Bcyc} \times (n + 1.5) - 33$ *1 | — | ns | Figures 26.11 to 26.15 |
| Access time from read strobe | $t_{OE}$ *2 | $t_{Bcyc} \times (n + 1) - 31$ *1 | — | ns | Figures 26.11 to 26.15 |
| Write strobe delay time 1 | $t_{WSD1}$ | $1/2t_{Bcyc} + 1$ | $1/2t_{Bcyc} + 18$ | ns | Figures 26.11 to 26.15 |
| Write data delay time 1 | $t_{WDD1}$ | — | 18 | ns | Figures 26.11 to 26.15 |
| Write data hold time 1 | $t_{WDH1}$ | 1 | 11 | ns | Figures 26.11 to 26.15 |
| Write data hold time | $t_{WRH}$ | 0 | — | ns | Figures 26.11 to 26.14 |
| $\overline{WAIT}$ setup time | $t_{WTS}$ | $1/2t_{Bcyc} + 17$ | — | ns | Figures 26.12 to 26.15 |
| $\overline{WAIT}$ hold time | $t_{WTH}$ | $1/2t_{Bcyc} + 7$ | — | ns | Figures 26.12 to 26.15 |

Notes: $t_{Bcyc}$ indicates external bus clock period (Bϕ = CK).

1. n denotes the number of wait cycles.
2. If the access time conditions are satisfied, the $t_{RDS1}$ condition does not need to be satisfied.

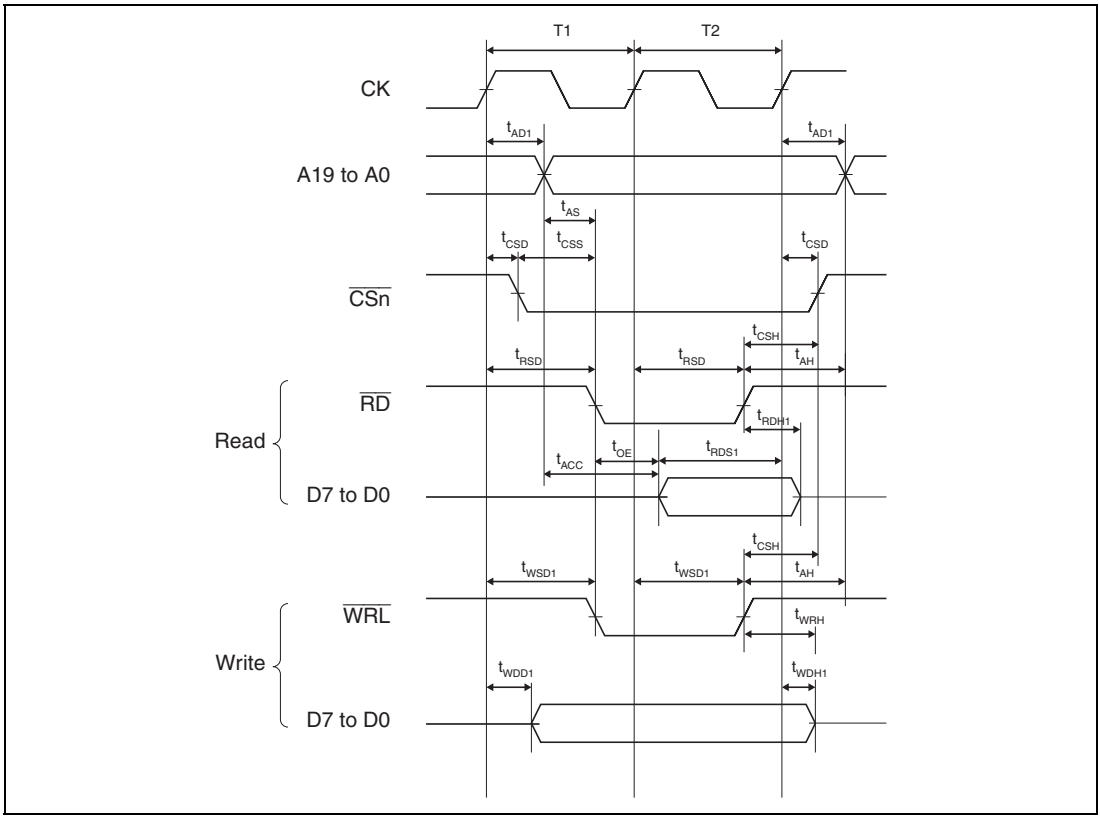



**Figure 26.11 Basic Bus Timing for Normal Space (No Wait)**

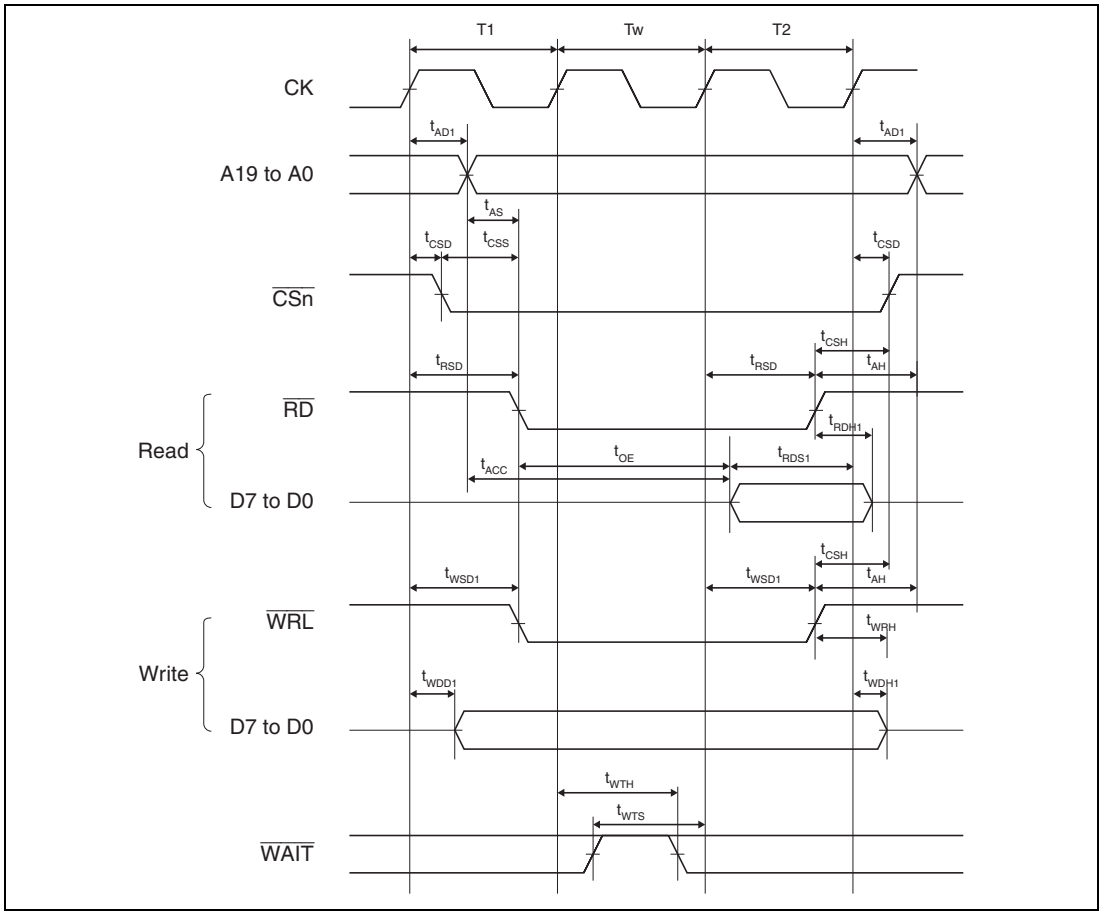



**Figure 26.12 Basic Bus Timing for Normal Space (One Software Wait Cycle)**

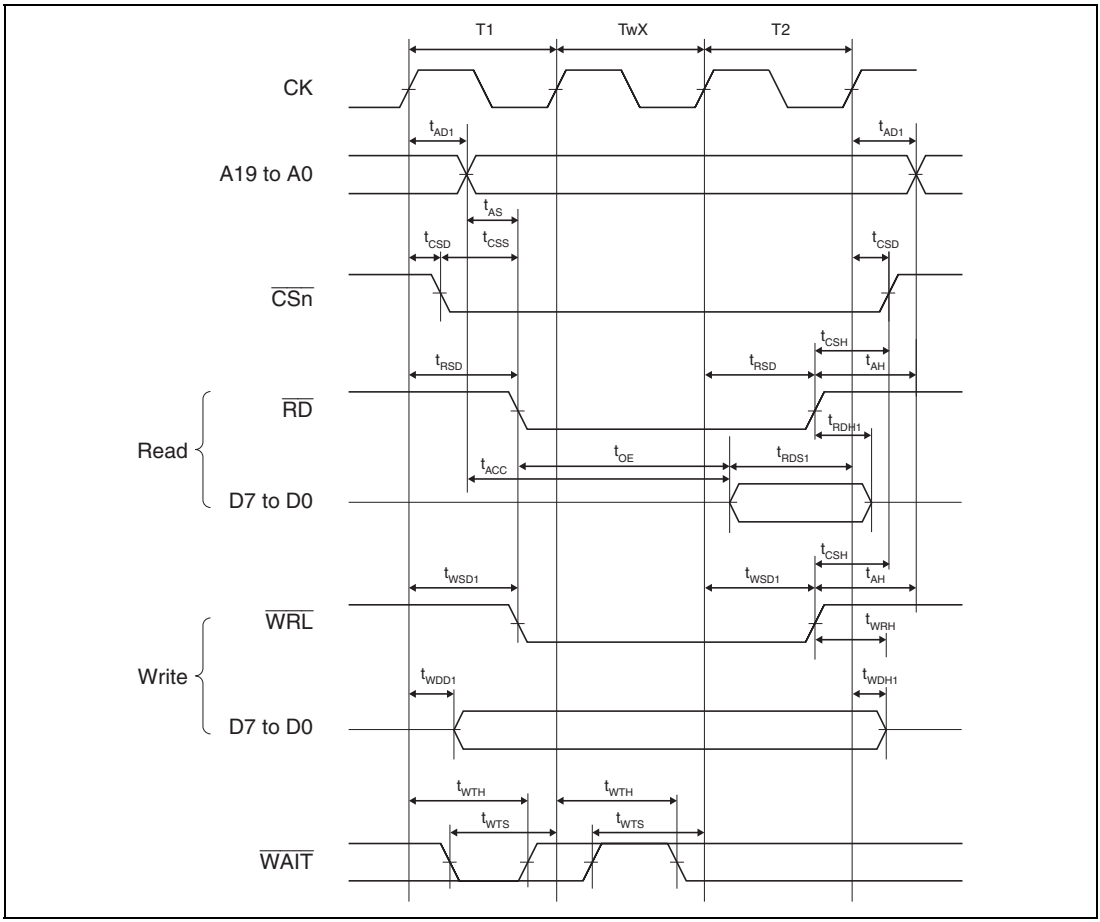



**Figure 26.13 Basic Bus Timing for Normal Space (One External Wait Cycle)**

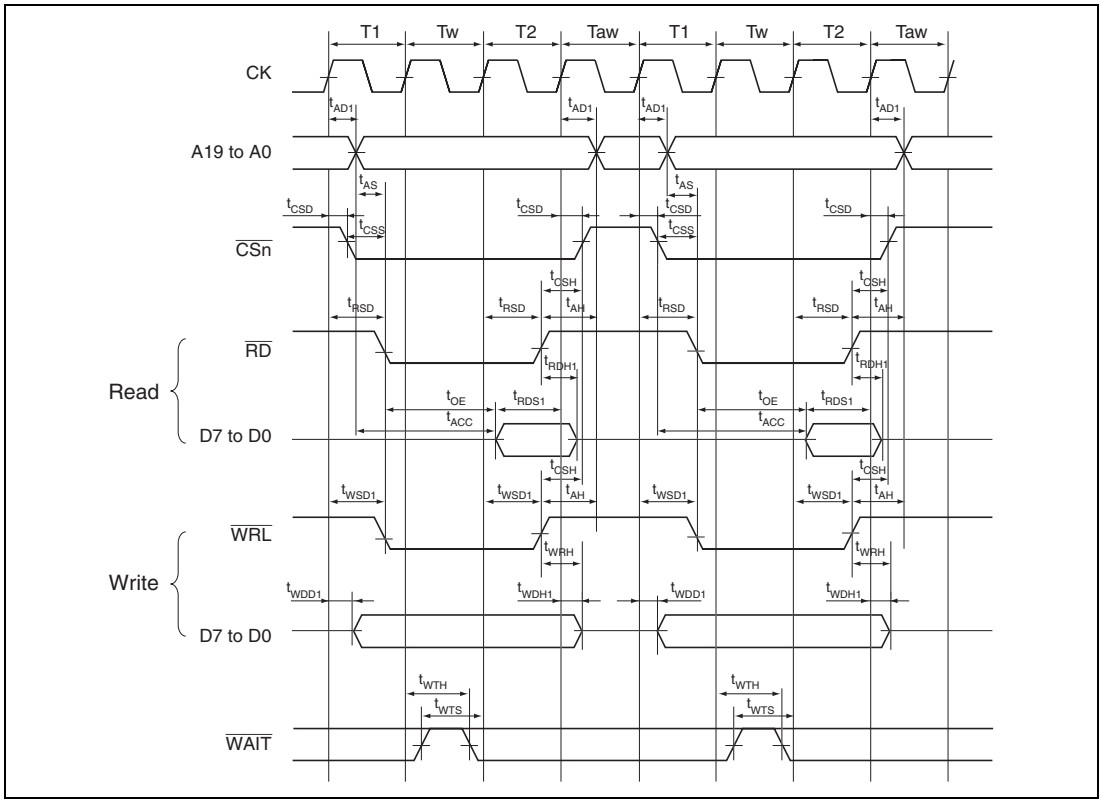



**Figure 26.14 Basic Bus Timing for Normal Space**
**(One Software Wait Cycle, External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)**

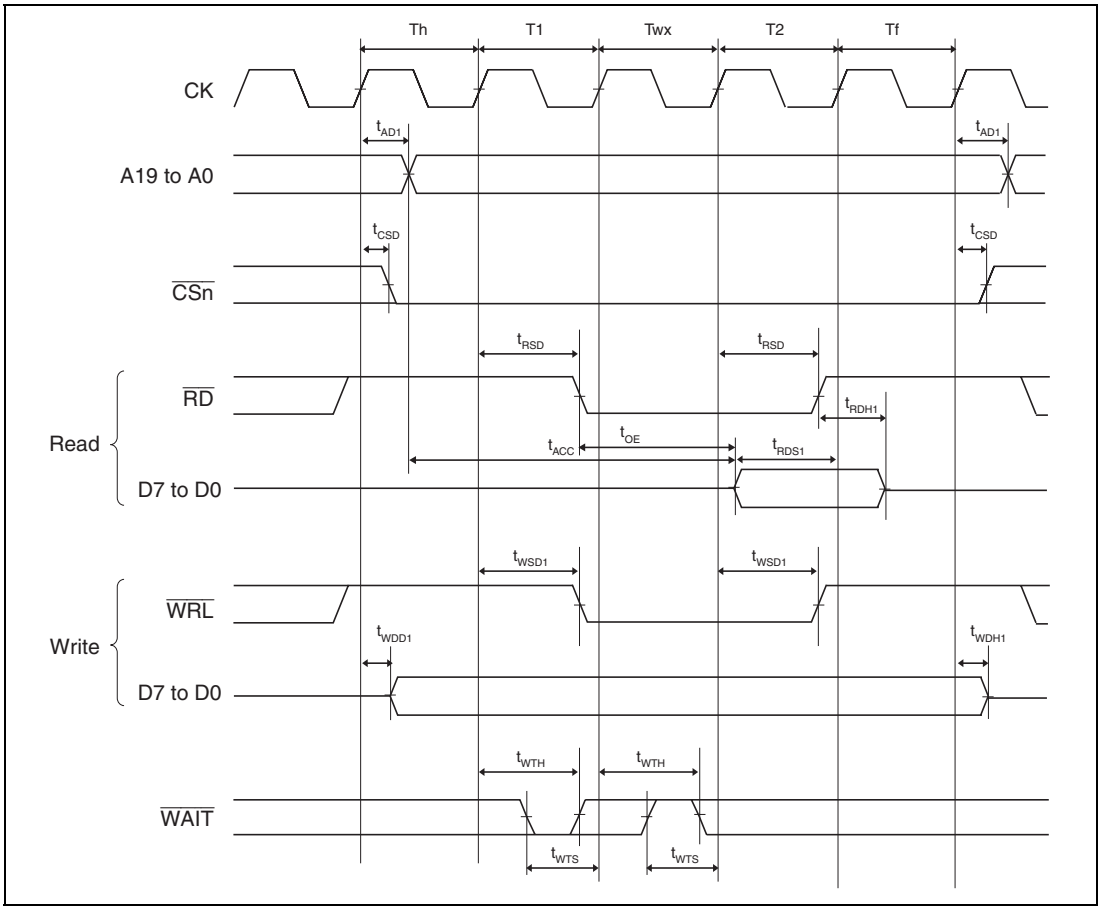



**Figure 26.15 CS Extended Bus Cycle for Normal Space (SW = 1 Cycle, HW = 1 Cycle, One External Wait Cycle)**

### 26.3.4 Multi Function Timer Pulse Unit 2 (MTU2) Timing

**Table 26.9 Multi Function Timer Pulse Unit 2 (MTU2) Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| Output compare output delay time | $t_{TOCD}$ | — | 50 | ns | Figure 26.16 |
| Input capture input setup time | $t_{TICS}$ | 20 | — | ns | |
| Input capture input pulse width (single edge) | $t_{TICW}$ | 1.5 | — | $t_{MPcyc}$ | |
| Input capture input pulse width (both edges) | $t_{TICW}$ | 2.5 | — | $t_{MPcyc}$ | |
| Timer input setup time | $t_{TCKS}$ | 20 | — | ns | Figure 26.17 |
| Timer clock pulse width (single edge) | $t_{TCKWH/L}$ | 1.5 | — | $t_{MPcyc}$ | |
| Timer clock pulse width (both edges) | $t_{TCKWH/L}$ | 2.5 | — | $t_{MPcyc}$ | |
| Timer clock pulse width (phase counting mode) | $t_{TCKWH/L}$ | 2.5 | — | $t_{MPcyc}$ | |

Note: $t_{MPcyc}$ indicates the MTU2 clock (MPϕ) cycle.

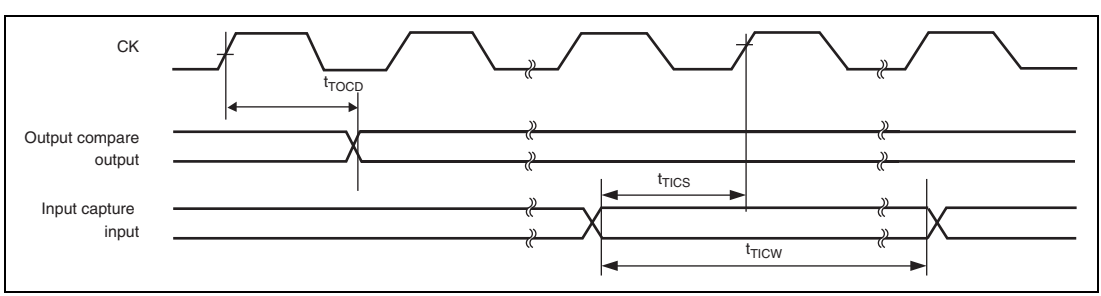



**Figure 26.16 MTU2 Input/Output Timing**

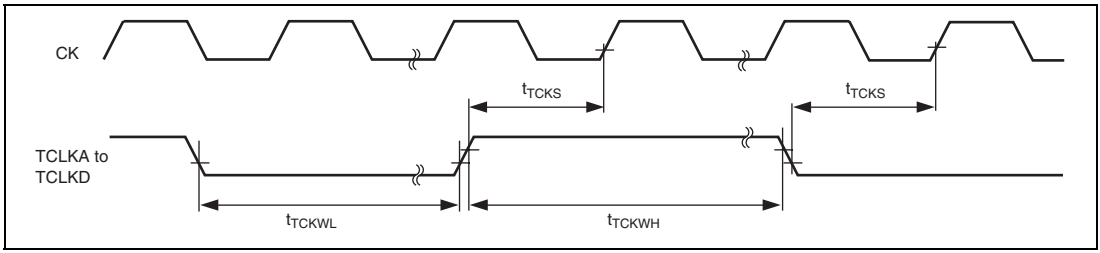



**Figure 26.17 MTU2 Clock Input Timing**

### 26.3.5 Multi Function Timer Pulse Unit 2S (MTU2S) Timing

**Table 26.10 Multi Function Timer Pulse Unit 2S (MTU2S) Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| Output compare output delay time | $t_{TOCD}$ | — | 50 | ns | Figure 26.18 |
| Input capture input setup time | $t_{TICS}$ | 20 | — | ns | |
| Input capture input pulse width (single edge) | $t_{TICW}$ | 1.5 | — | $t_{MIcyc}$ | |
| Input capture input pulse width (both edges) | $t_{TICW}$ | 2.5 | — | $t_{MIcyc}$ | |

Note: $t_{MIcyc}$ indicates the MTU2S clock (MIϕ) cycle.

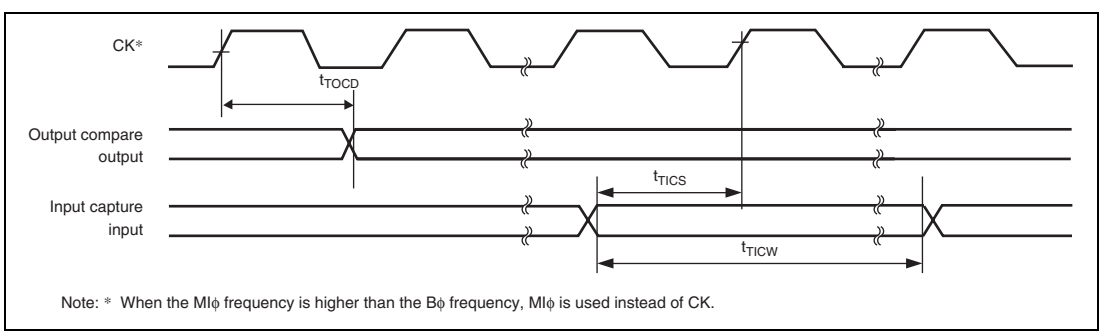



**Figure 26.18 MTU2S Input/Output Timing**

### 26.3.6 I/O Port Timing

**Table 26.11 I/O Port Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| Port output data delay time | $t_{PWD}$ | — | 50 | ns | Figure 26.19 |
| Port input hold time | $t_{PRH}$ | 20 | — | ns | |
| Port input setup time | $t_{PRS}$ | 20 | — | ns | |

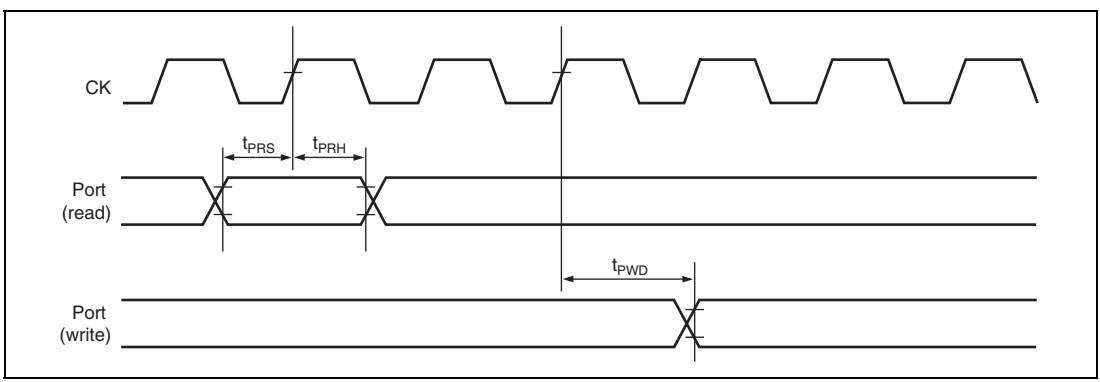



**Figure 26.19 I/O Port Input/Output Timing**

### 26.3.7 Watchdog Timer (WDT) Timing

**Table 26.12 Watchdog Timer (WDT) Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| $\overline{\text{WDTOVF}}$ delay time | $t_{WOVD}$ | — | 50 | ns | Figure 26.20 |

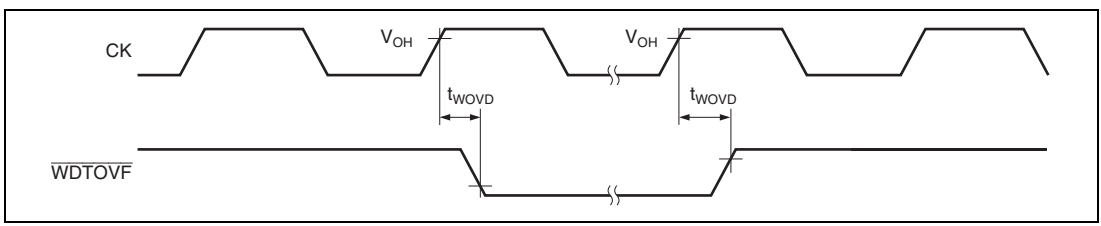



**Figure 26.20 WDT Timing**

### 26.3.8 Serial Communication Interface (SCI) Timing

**Table 26.13 Serial Communication Interface (SCI) Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|
| Input clock cycle (asynchronous) | | $t_{scyc}$ | 4 | — | $t_{pcyc}$ | Figure 26.21 |
| Input clock cycle (clock synchronous) | | $t_{scyc}$ | 6 | — | $t_{pcyc}$ | |
| Input clock pulse width | | $t_{sckw}$ | 0.4 | 0.6 | $t_{scyc}$ | |
| Input clock rise time | | $t_{sckr}$ | — | 1.5 | $t_{pcyc}$ | |
| Input clock fall time | | $t_{sckf}$ | — | 1.5 | $t_{pcyc}$ | |
| Transmit data delay time | Asynchronous | $t_{TXD}$ | — | 4 $t_{pcyc}$ + 10 | ns | Figure 26.22 |
| Receive data setup time | | $t_{RXS}$ | 4 $t_{pcyc}$ | — | ns | |
| Receive data hold time | | $t_{RXH}$ | 4 $t_{pcyc}$ | — | ns | |
| Transmit data delay time | Clock synchronous | $t_{TXD}$ | — | 3 $t_{pcyc}$ + 10 | ns | |
| Receive data setup time | | $t_{RXS}$ | 2 $t_{pcyc}$ + 50 | — | ns | |
| Receive data hold time | | $t_{RXH}$ | 2 $t_{pcyc}$ | — | ns | |

Note: $t_{pcyc}$ indicates the peripheral clock (Pϕ) cycle.

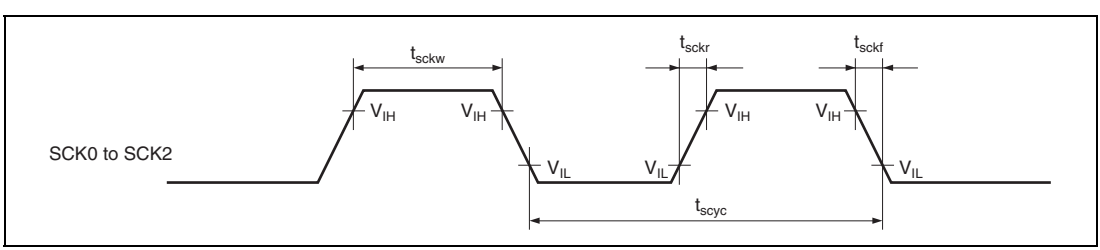



**Figure 26.21 Input Clock Timing**

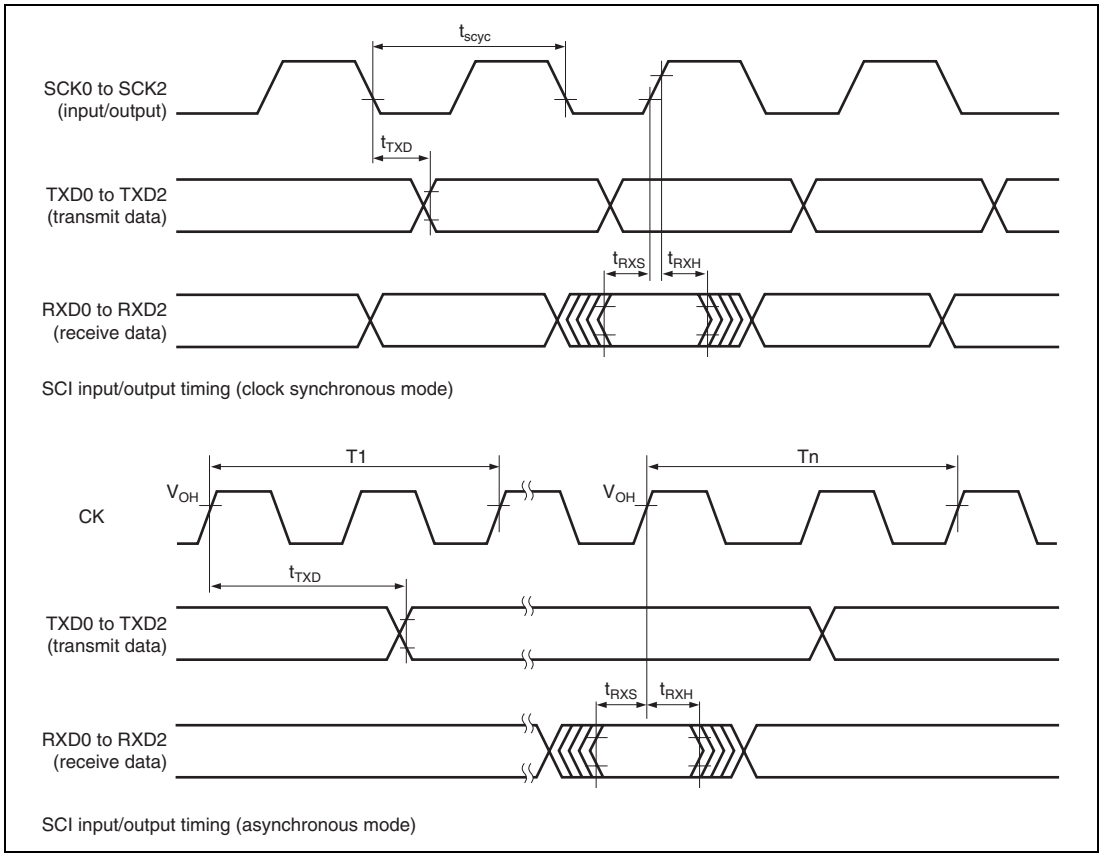



**Figure 26.22 SCI Input/Output Timing**

### 26.3.9 Synchronous Serial Communication Unit (SSU) Timing

**Table 26.14 Synchronous Serial Communication Unit (SSU) Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|
| Clock cycle | Master | $t_{SUcyc}$ | 4 | 256 | $t_{pcyc}$ | Figures 26.23 to 26.26 |
| | Slave | | 4 | 256 | | |
| Clock high pulse width | Master | $t_{HI}$ | 60 | — | ns | |
| | Slave | | 60 | — | | |
| Clock low pulse width | Master | $t_{LO}$ | 60 | — | ns | |
| | Slave | | 60 | — | | |
| Clock rise time | | $t_{RISE}$ | — | 20 | ns | |
| Clock fall time | | $t_{FALL}$ | — | 20 | ns | |
| Data input setup time | Master | $t_{SU}$ | 30 | — | ns | |
| | Slave | | 30 | — | | |
| Data input hold time | Master | $t_H$ | 10 | — | ns | |
| | Slave | | 10 | — | | |
| $\overline{SCS}$ setup time | Master | $t_{LEAD}$ | 1.5 | — | $t_{pcyc}$ | |
| | Slave | | 1.5 | — | | |
| $\overline{SCS}$ hold time | Master | $t_{LAG}$ | 1.5 | — | $t_{pcyc}$ | |
| | Slave | | 1.5 | — | | |
| Data output delay time | Master | $t_{OD}$ | — | 40 | ns | |
| | Slave | | — | 40 | | |
| Data output hold time | Master | $t_{OH}$ | 30 | — | ns | |
| | Slave | | 30 | — | | |
| Continuous transmission delay time | Master | $t_{TD}$ | 1.5 | — | $t_{pcyc}$ | |
| | Slave | | 1.5 | — | | |
| Slave access time | | $t_{SA}$ | — | 1 | $t_{pcyc}$ | Figures 26.25, 26.26 |
| Slave out release time | | $t_{REL}$ | — | 1 | $t_{pcyc}$ | |

Note: $t_{pcyc}$ indicates the peripheral clock (Pϕ) cycle.

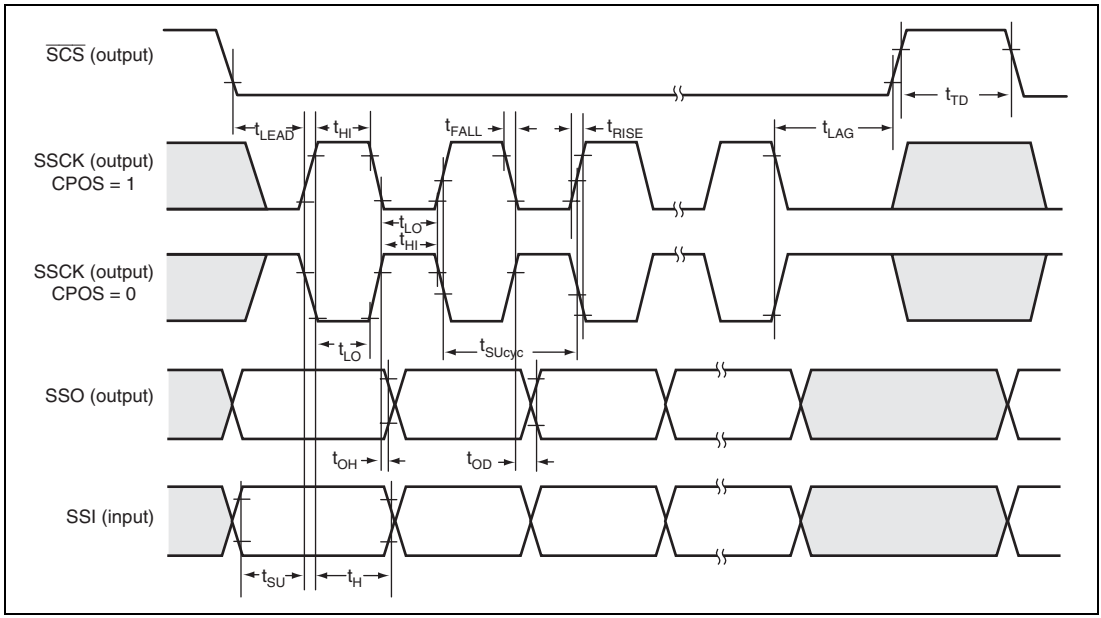



**Figure 26.23 SSU Timing (Master, CPHS = 1)**

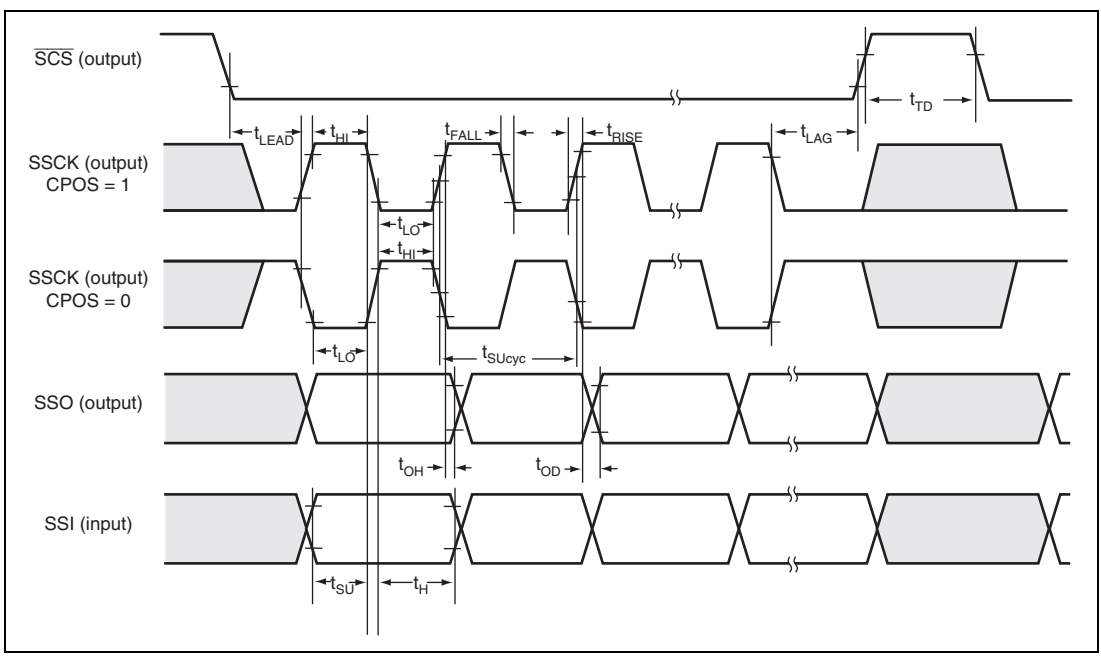



**Figure 26.24 SSU Timing (Master, CPHS = 0)**

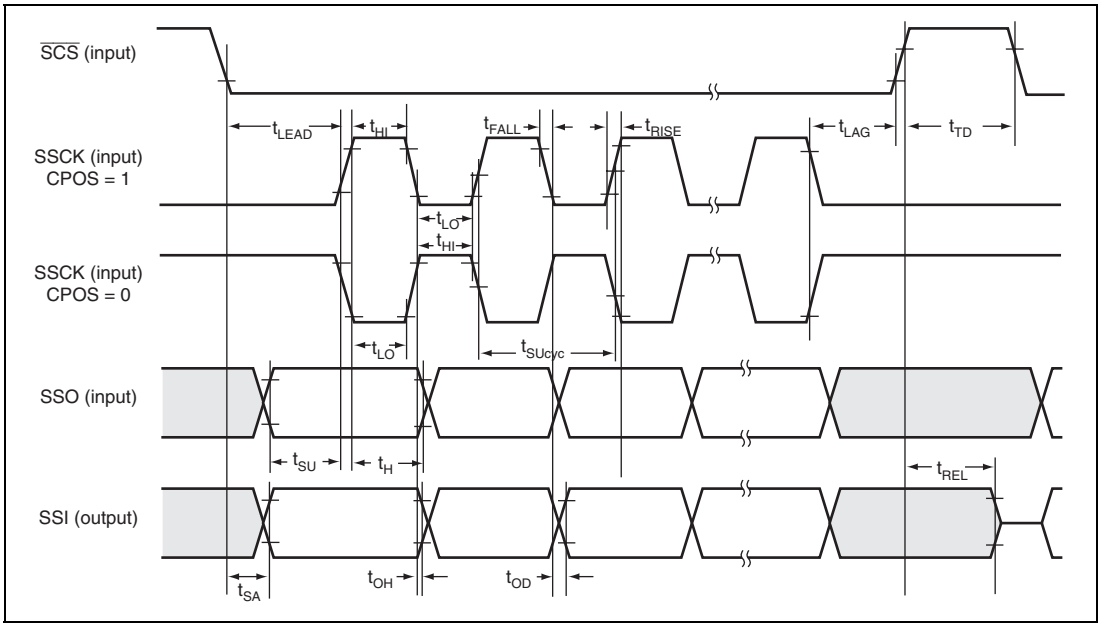



**Figure 26.25 SSU Timing (Slave, CPHS = 1)**

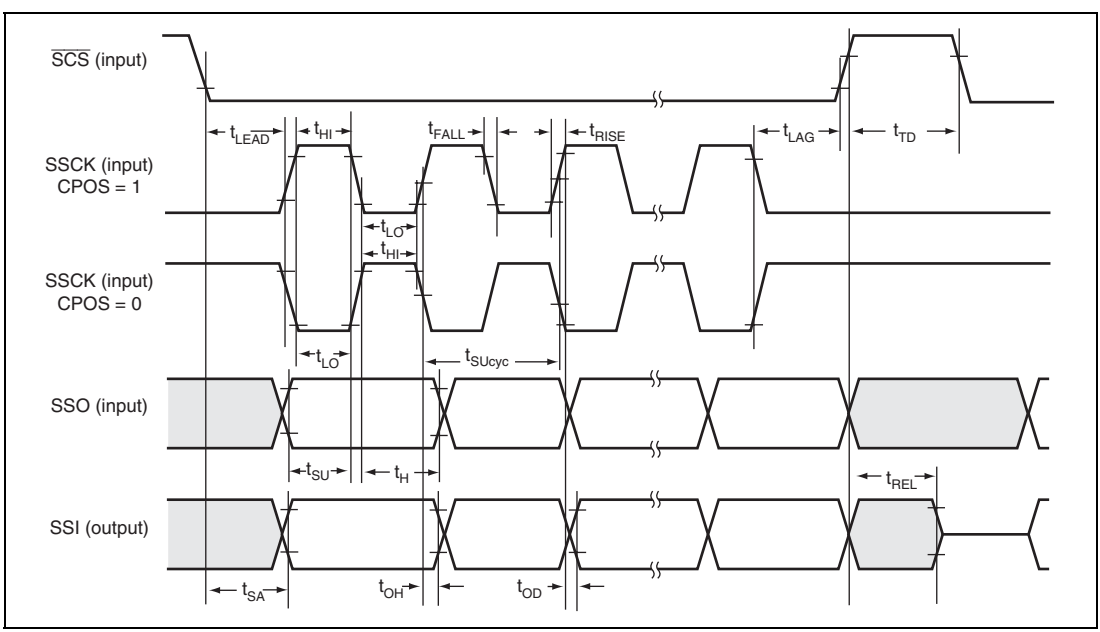



**Figure 26.26 SSU Timing (Slave, CPHS = 0)**

### 26.3.10 Controller Area Network (RCAN-ET) Timing

Table 26.15 shows RCAN-ET timing.

**Table 26.15 Controller Area Network (RCAN-ET) Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V,
$AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$,
$V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
Ta = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| Transmit data delay time | $t_{CTxD}$ | — | 100 | ns | Figure 26.27 |
| Receive data setup time | $t_{CRxS}$ | 100 | — | ns | |
| Receive data hold time | $t_{CRxH}$ | 100 | — | ns | |

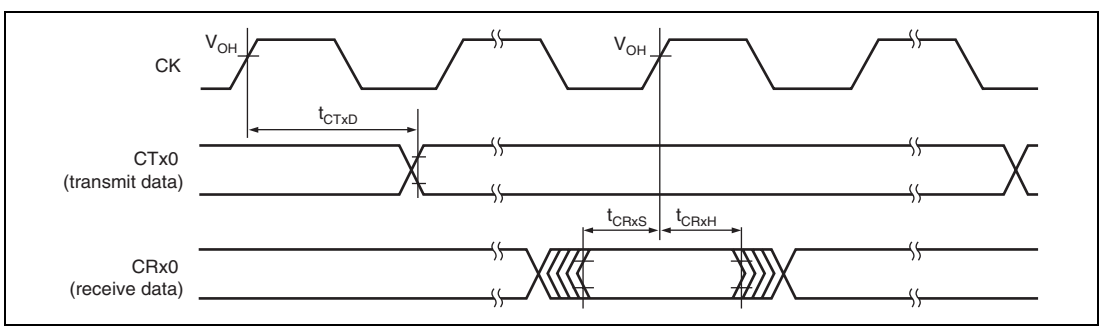



**Figure 26.27 RCAN-ET Input/Output Timing**

### 26.3.11 Port Output Enable (POE) Timing

**Table 26.16 Port Output Enable (POE) Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| $\overline{POE}$ input setup time | $t_{POES}$ | 50 | — | ns | Figure 26.28 |
| $\overline{POE}$ input pulse width | $t_{POEW}$ | 1.5 | — | $t_{pcyc}$ | |

Note: $t_{pcyc}$ indicates the peripheral clock (Pϕ) cycle.

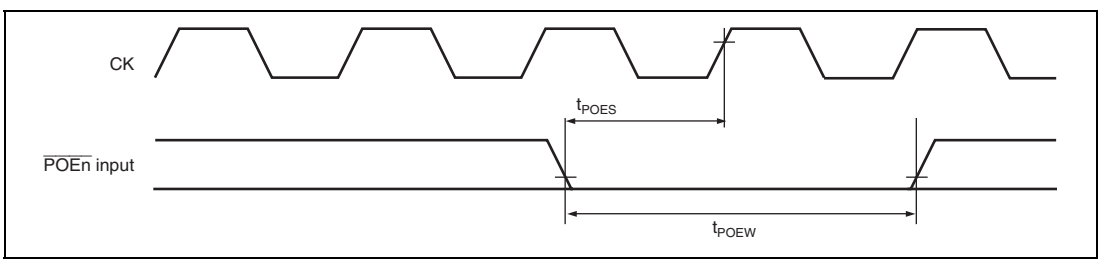



**Figure 26.28 $\overline{POE}$ Input Timing**

### 26.3.12 $I^2C$ Bus Interface 2 ($I^2C2$) Timing

**Table 26.17 $I^2C$ Bus Interface 2 ($I^2C2$) Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Typ. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|
| SCL input cycle time | $t_{SCL}$ | 12 $t_{pcyc}$ + 600 | — | — | ns | Figure 26.29 |
| SCL input high pulse width | $t_{SCLH}$ | 3 $t_{pcyc}$ + 300 | — | — | ns | |
| SCL input low pulse width | $t_{SCLL}$ | 5 $t_{pcyc}$ + 300 | — | — | ns | |
| SCL and SDA input fall time | $t_{Sf}$ | — | — | 300 | ns | |
| SCL and SDA input spike pulse removal time | $t_{SP}$ | — | — | 1 $t_{pcyc}$ | ns | |
| SDA input bus free time | $t_{BUF}$ | 5 | — | — | $t_{pcyc}$ | |
| Start condition input hold time | $t_{STAH}$ | 3 | — | — | $t_{pcyc}$ | |
| Repeated start condition input setup time | $t_{STAS}$ | 3 | — | — | $t_{pcyc}$ | |
| Halt condition input setup time | $t_{STOS}$ | 3 | — | — | $t_{pcyc}$ | |
| Data input setup time | $t_{SDAS}$ | 1 $t_{pcyc}$ + 20 | — | — | ns | |
| Data input hold time | $t_{SDAH}$ | 0 | — | — | ns | |
| SCL and SDA capacity load | $C_b$ | 0 | — | 400 | pF | |
| SCL and SDA output fall time | $t_{Sf}$ | — | — | 250 | ns | |

Note: $t_{pcyc}$ indicates the peripheral clock (Pϕ) cycle.

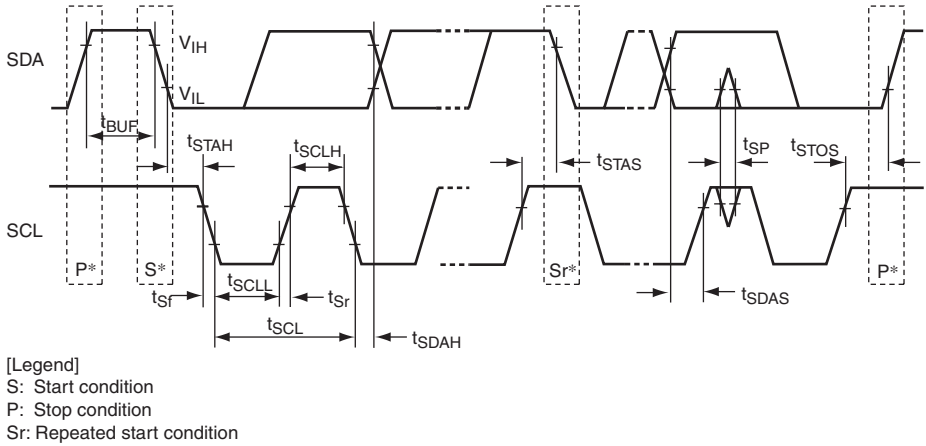



**Figure 26.29 I²C2 Input/Output Timing**

### 26.3.13 UBC Trigger Timing

**Table 26.18 UBC Trigger Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|
| $\overline{\text{UBCTRG}}$ delay time | $t_{UBCTGD}$ | — | 150 | ns | Figure 26.30 |

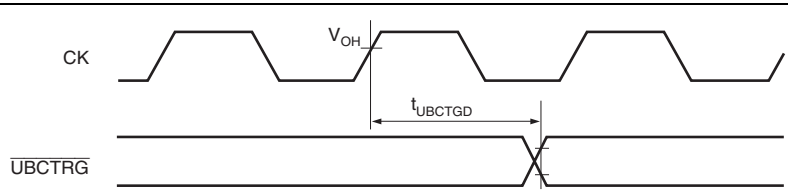



**Figure 26.30 UBC Trigger Timing**

### 26.3.14 A/D Converter Timing

**Table 26.19 A/D Converter Timing**

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Symbol | Min. | Typ. | Max. | Unit | Figure |
|---|---|---|---|---|---|---|
| External trigger input start delay time | $t_{TRGS}$ | 25 | — | — | ns | Figure 26.31 |

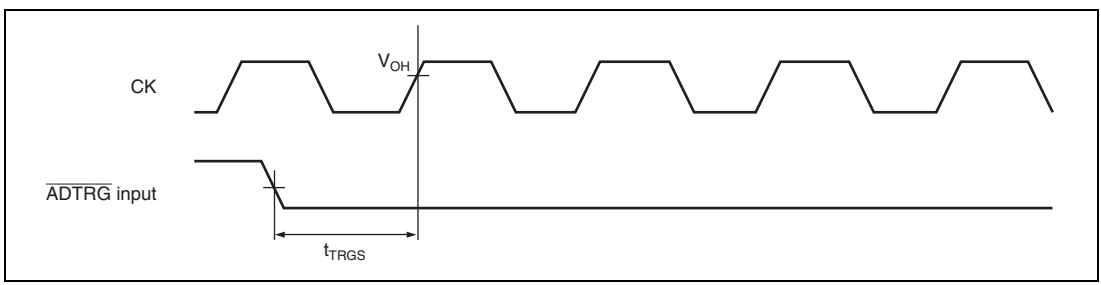
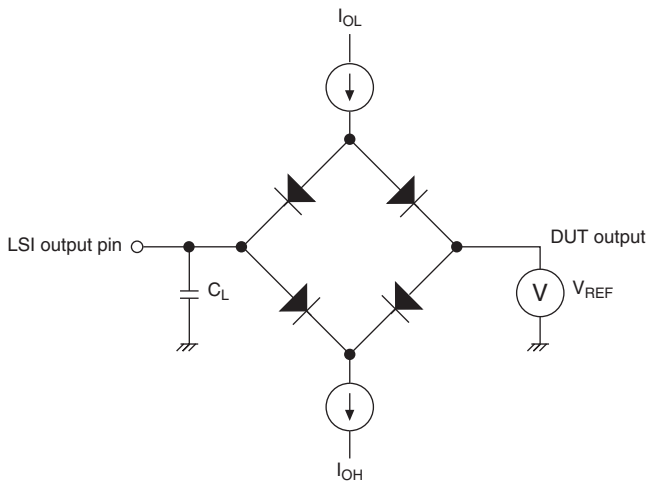



**Figure 26.31 External Trigger Input Timing**

### 26.3.15 AC Characteristics Measurement Conditions

- Input signal level: $V_{IL}$ (Max.)/$V_{IH}$ (Min.)
- Output signal reference level: High level: 2.0 V, Low level: 0.8 V




Notes: 1. $C_L$ is the total value that includes the capacitance of measurement tools. Each pin is set as follows:
20pF: CK
30pF: All other output pins
2. Test conditions include $I_{OL}$ = 1.6 mA and $I_{OH}$ = -200 μA.

**Figure 26.32 Output Load Circuit**

## 26.4 A/D Converter Characteristics

### Table 26.20 A/D Converter Characteristics

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

| Item | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|
| Resolution | 12 | 12 | 12 | bit |
| A/D conversion time | 1.25*[1] | — | — | μs |
| Analog input capacitance | — | — | 5 | pF |
| Permitted analog signal source impedance | — | — | 3 | kΩ |
| Non-linear error | — | — | ±4*[2] | LSB |
| Offset error | — | — | ±7.5*[2] | LSB |
| Full-scale error | — | — | ±7.5*[2] | LSB |
| Quantization error | — | — | ±0.5*[2] | LSB |
| Absolute error*[3] | — | — | ±8 | LSB |

Notes: 1. Conversion time per channel when the sample-and-hold circuit is not used and the A/D clock operates at 40 MHz.
2. Reference value.
3. Guaranteed range from $AV_{refl}$ + 0.25 V to $AV_{refh}$ – 0.25 V.

## 26.5 Flash Memory Characteristics

### Table 26.21 Flash Memory Characteristics

Conditions: $V_{CC}$ = 3.0 V to 3.6 V or 4.0 V to 5.5 V, $AV_{CC}$ = 4.5 V to 5.5 V, $AV_{refh}$ = 4.5 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 0 V,
$T_a$ = –20°C to +85°C (consumer applications),
$T_a$ = –40°C to +85°C (industrial applications)

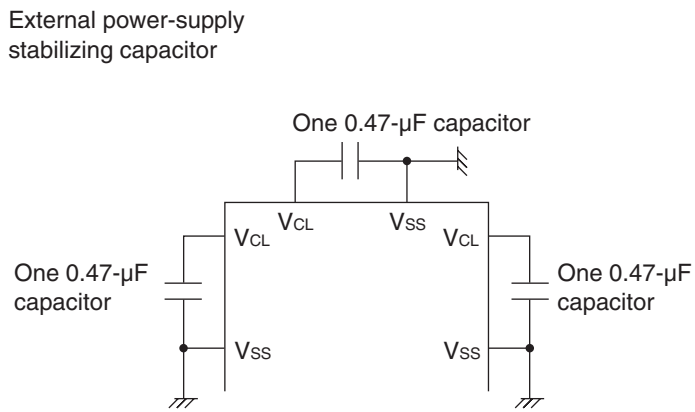
| Item | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Programming time*1*2*4 | $t_P$ | — | 1 | 20 | ms/128 bytes |
| Erase time*1*2*4 | $t_E$ | — | 40 | 260 | ms/4 Kbyte block |
| | | — | 300 | 1500 | ms/32 Kbyte block |
| | | — | 600 | 3000 | ms/64 Kbyte block |
| Programming time (total)*1*2*4 | $\Sigma t_P$ | — | 2.3 | 12 | s/256 Kbytes |
| | | — | 1.1 | 6 | s/128 Kbytes |
| Erase time (total)*1*2*4 | $\Sigma t_E$ | — | 2.3 | 12 | s/256 Kbytes |
| | | — | 1.1 | 6 | s/128 Kbytes |
| Programming and erase time (total)*1*2*4 | $\Sigma t_{PE}$ | — | 4.6 | 24 | s/256 Kbytes |
| | | — | 2.2 | 12 | s/128 Kbytes |
| Reprogramming count | $N_{WEC}$ | 500*3 | — | — | Times |

Notes: 1. Programming and erase time vary depending on the data.
2. Programming and erase time do not include data transfer time.
3. The minimum number of times for which all characteristics are guaranteed after reprogramming (guaranteed for once to the minimum number of reprogramming times).
4. These characteristics only apply when reprogramming is performed within the range of minimum number of times.

## 26.6 Usage Note

### 26.6.1 Notes on Connecting $V_{CL}$ Capacitor

This LSI includes an internal step-down circuit to automatically reduce the internal power supply voltage to an appropriate level. Between this internal stepped-down power supply ($V_{CL}$ pin) and the $V_{SS}$ pin, a capacitor (0.47 μF) for stabilizing the internal voltage needs to be connected. Connection of the external capacitor is shown in figure 26.33. The external capacitor should be located near the pin. Do not apply any power supply voltage to the $V_{CL}$ pin.




Note: Do not apply any power supply voltage to the $V_{CL}$ pin.
Use multilayer ceramic capacitors (one 0.47-μF capacitor for each $V_{CL}$ pin), which should be located near the pin.

**Figure 26.33 Connection of $V_{CL}$ Capacitor**

# Appendix

## A. Pin States

Pin initial states differ according to MCU operating modes. Refer to section 20, Pin Function Controller (PFC), for details.

### Table A.1 Pin States (SH7131/SH7136)

| Pin Function | | Pin State | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Reset State | | Power-Down State | | | | |
| Type | Pin Name | Power-On | Manual | Deep Software Standby*4 | Software Standby*4 | Sleep | Oscillation Stop Detected | POE Function Used |
| Clock | XTAL | O | O | L | L | O | O | O |
| | EXTAL | I | I | Z | I | I | I | I |
| System control | $\overline{\text{RES}}$ | I | I | I | I | I | I | I |
| | $\overline{\text{MRES}}$ | Z | I | Z | Z | I | Z | I |
| | $\overline{\text{WDTOVF}}$ | O*2 | O | O | O | O | O | O |
| Operating mode control | MD1 | I | I | I | I | I | I | I |
| | $\overline{\text{ASEMD0}}$ | I*3 | I*3 | I*3 | I*3 | I*3 | I*3 | I*3 |
| | FWE | I | I | I | I | I | I | I |
| Interrupt | NMI | I | I | I | I | I | I | I |
| | IRQ0 to IRQ3 | Z | I | Z | I | I | I | I |
| | $\overline{\text{IRQOUT}}$ | Z | O | Z | Z | O | Z | O |
| MTU2 | TCLKA to TCLKD | Z | I | Z | Z | I | I | I |
| | TIOC0A to TIOC0D | Z | I/O | Z | K*1 | I/O | I/O | Z |
| | TIOC1A, TIOC1B | Z | I/O | Z | K*1 | I/O | I/O | I/O |
| | TIOC2A, TIOC2B | Z | I/O | Z | K*1 | I/O | I/O | I/O |
| | TIOC3A, TIOC3C | Z | I/O | Z | K*1 | I/O | I/O | I/O |
| | TIOC3B, TIOC3D | Z | I/O | Z | Z | I/O | Z | Z |

| Pin Function | | Pin State | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Reset State | | Power-Down State | | | | |
| Type | Pin Name | Power-On | Manual | Deep Software Standby*4 | Software Standby*4 | Sleep | Oscillation Stop Detected | POE Function Used |
| MTU2 | TIOC4A to TIOC4D | Z | I/O | Z | Z | I/O | Z | Z |
| | TIC5U, TIC5V | Z | I | Z | Z | I | I | I |
| MTU2S | TIOC3BS, TIOC3DS | Z | I/O | Z | Z | I/O | Z | Z |
| | TIOC4AS to TIOC4DS | Z | I/O | Z | Z | I/O | Z | Z |
| | TIC5US, TIC5VS | Z | I | Z | Z | I | I | I |
| POE | $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, $\overline{\text{POE8}}$ | Z | I | Z | Z | I | I | I |
| SCI | SCK0 to SCK2 | Z | I/O | Z | Z | I/O | I/O | I/O |
| | RXD0 to RXD2 | Z | I | Z | Z | I | I | I |
| | TXD0 to TXD2 | Z | O | Z | O*1 | O | O | O |
| SSU | SSCK | Z | I/O | Z | Z | I/O | I/O | I/O |
| | $\overline{\text{SCS}}$ | Z | I/O | Z | Z | I/O | I/O | I/O |
| | SSI | Z | I/O | Z | Z | I/O | I/O | I/O |
| | SSO | Z | I/O | Z | Z | I/O | I/O | I/O |
| $I^2C2$ | SCL | Z | I/O | Z | Z | I/O | I/O | I/O |
| | SDA | Z | I/O | Z | Z | I/O | I/O | I/O |
| UBC | $\overline{\text{UBCTRG}}$*4 | Z | O | Z | O*1 | O | O | O |
| RCAN-ET | CTx0 | Z | O | Z | O*1 | O | O | O |
| | CRx0 | Z | I | Z | Z | I | I | I |
| A/D Converter | AN0 to AN3, AN8 to AN15 | Z | I | Z | Z | I | I | I |
| | $\overline{\text{ADTRG}}$ | Z | I | Z | Z | I | I | I |

| Pin Function | | Pin State | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Reset State | | Power-Down State | | | | |
| Type | Pin Name | Power-On | Manual | Deep Software Standby | Software Standby | Sleep | Oscillation Stop Detected | POE Function Used |
| I/O Port | PA0 to PA15 | Z | I/O | Z | K※1 | I/O | I/O | I/O |
| | PB2 to PB7 | Z | I/O | Z | K※1 | I/O | I/O | I/O |
| | PE0 to PE3 | Z | I/O | Z | K※1 | I/O | I/O | Z |
| | PE4 to PE8, PE10 | Z | I/O | Z | K※1 | I/O | I/O | I/O |
| | PE9, PE11 to PE15 | Z | I/O | Z | Z | I/O | Z | Z |
| | PE16 to PE21 | Z | I/O | Z | Z | I/O | Z | Z |
| | PF0 to PF3, PF8 to PF15 | Z | I | Z | Z | I | I | I |

[Legend]

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High-impedance

K: Input pins become high-impedance, and output pins retain their state.

Notes:
1. Output pins become high-impedance when the HIZ bit in standby control register 6 (STBCR6) is set to 1.
2. Becomes input during a power-on reset. Pull-up to prevent erroneous operation. Pull-down with a resistance of at least 1 MΩ as required.
3. Pulled-up inside the LSI when there is no input.
4. SH7136 only.

## Table A.2 Pin States (SH7132/SH7137)

| Pin Function | | Pin State | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Reset State | | | | Power-Down State | | | | | |
| | | Power-On | | | | | | | | | |
| Type | Pin Name | Expansion without ROM 8 bits | Expansion with ROM | Single-Chip | Manual | Deep Software Standby | Software Standby | Sleep | Bus Master-ship Release | Oscillation Stop Detected | POE Function Used |
| Clock | CK | O | | Z | O | Z | H*1 | O | O | O | O |
| | XTAL | O | | | O | L | L | O | O | O | O |
| | EXTAL | I | | | I | Z | I | I | I | I | I |
| System control | $\overline{\text{RES}}$ | I | | | I | I | I | I | I | I | I |
| | $\overline{\text{MRES}}$ | Z | | | I | Z | Z | I | I | Z | I |
| | $\overline{\text{WDTOVF}}$ | O*3 | | | O | O | O | O | O | O | O |
| | $\overline{\text{BREQ}}$ | Z | | | I | Z | Z | I | I | I | I |
| | $\overline{\text{BACK}}$ | Z | | | O | Z | Z | O | L | O | O |
| Operating mode control | MD0, MD1 | I | | | I | I | I | I | I | I | I |
| | $\overline{\text{ASEMD0}}$ | I*4 | | | I*4 | I*4 | I*4 | I*4 | I*4 | I*4 | I*4 |
| | FWE | I | | | I | I | I | I | I | I | I |
| Interrupt | NMI | I | | | I | I | I | I | I | I | I |
| | IRQ0 to IRQ3 | Z | | | I | Z | I | I | I | I | I |
| | $\overline{\text{IRQOUT}}$ | Z | | | O | Z | Z | O | O | Z | O |
| Address bus | A0 to A17 | O | Z | | O | Z | Z*2 | O | Z | O | O |
| | A18, A19 | Z | | | O | Z | Z*2 | O | Z | O | O |
| Data bus | D0 to D7 | Z | | | I/O | Z | Z | I/O | Z | I/O | I/O |
| Bus control | $\overline{\text{WAIT}}$ | Z | | | I | Z | Z | I | Z | I | I |
| | $\overline{\text{CS0}}$ (PE10) | H | Z | | O | Z | Z*2 | O | Z | O | O |
| | $\overline{\text{CS0}}$ (PE17), $\overline{\text{CS1}}$ (PE18) | Z | | | O | Z | Z*2 | O | Z | O | O |
| | $\overline{\text{RD}}$ (PA6) | H | Z | | O | Z | Z*2 | O | Z | O | O |
| | $\overline{\text{RD}}$ (PE19) | Z | | | O | Z | Z*2 | O | Z | O | O |

| Pin Function | | Pin State | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Reset State | | | | Power-Down State | | | | | |
| | | Power-On | | | | | | | | | |
| Type | Pin Name | Expansion without ROM 8 bits | Expansion with ROM | Single-Chip | Manual | Deep Software Standby | Software Standby | Sleep | Bus Master-ship Release | Oscillation Stop Detected | POE Function Used |
| Bus control | $\overline{\text{WRL}}$ (PA8) | H | Z | | O | Z | Z※2 | O | Z | O | O |
| | $\overline{\text{WRL}}$ (PE21) | Z | | | O | Z | Z※2 | O | Z | O | O |
| MTU2 | TCLKA to TCLKD | Z | | | I | Z | Z | I | I | I | I |
| | TIOC0A to TIOC0D | Z | | | I/O | Z | K※1 | I/O | I/O | I/O | Z |
| | TIOC1A, TIOC1B | Z | | | I/O | Z | K※1 | I/O | I/O | I/O | I/O |
| | TIOC2A, TIOC2B | Z | | | I/O | Z | K※1 | I/O | I/O | I/O | I/O |
| | TIOC3A, TIOC3C | Z | | | I/O | Z | K※1 | I/O | I/O | I/O | I/O |
| | TIOC3B, TIOC3D | Z | | | I/O | Z | Z | I/O | I/O | Z | Z |
| | TIOC4A to TIOC4D | Z | | | I/O | Z | Z | I/O | I/O | Z | Z |
| | TIC5U, TIC5V, TIC5W | Z | | | I | Z | Z | I | I | I | I |
| MTU2S | TIOC3BS, TIOC3DS | Z | | | I/O | Z | Z | I/O | I/O | Z | Z |
| | TIOC4AS to TIOC4DS | Z | | | I/O | Z | Z | I/O | I/O | Z | Z |
| | TIC5US, TIC5VS, TIC5WS | Z | | | I | Z | Z | I | I | I | I |
| POE | $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, $\overline{\text{POE8}}$ (PA9) | Z | | | I | Z | Z | I | I | I | I |

| Pin Function | | Pin State | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Reset State | | | | Power-Down State | | | | | |
| | | Power-On | | | | | | | | | |
| Type | Pin Name | Expansion without ROM 8 bits | Expansion with ROM | Single-Chip | Manual | Deep Software Standby | Software Standby | Sleep | Bus Master-ship Release | Oscillation Stop Detected | POE Function Used |
| SCI | SCK0 to SCK2 | Z | | | I/O | Z | Z | I/O | I/O | I/O | I/O |
| | RXD0 to RXD2 | Z | | | I | Z | Z | I | I | I | I |
| | TXD0 to TXD2 | Z | | | O | Z | O*1 | O | O | O | O |
| SSU | SSCK | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O | I/O |
| | $\overline{\text{SCS}}$ | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O | I/O |
| | SSI | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O | I/O |
| | SSO | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O | I/O |
| $I^2C2$ | SCL | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O | I/O |
| | SDA | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O | I/O |
| UBC | $\overline{\text{UBCTRG}}$*5 | Z | | | O | Z | O*1 | O | O | O | O |
| RCAN-ET | CTx0 | Z | Z | O*1 | O | O | O | O | RCAN-ET | CTx0 | Z |
| | CRx0 | Z | Z | Z | I | I | I | I | | CRx0 | Z |
| A/D Converter | AN0 to AN15 | Z | | | I | Z | Z | I | I | I | I |
| | $\overline{\text{ADTRG}}$ | Z | | | I | Z | Z | I | I | I | I |
| I/O Port | PA0 to PA15 | Z | | | I/O | Z | K*1 | I/O | I/O | I/O | I/O |
| | PB0 to PB7 | Z | | | I/O | Z | K*1 | I/O | I/O | I/O | I/O |
| | PD0 to PD10 | Z | | | I/O | Z | K*1 | I/O | I/O | I/O | I/O |
| | PE0 to PE3 | Z | | | I/O | Z | K*1 | I/O | I/O | I/O | Z |
| | PE4 to PE8, PE10 | Z | | | I/O | Z | K*1 | I/O | I/O | I/O | I/O |
| | PE9, PE11 to PE15 | Z | | | I/O | Z | Z | I/O | I/O | Z | Z |
| | PE16 to PE21 | Z | | | I/O | Z | Z | I/O | I/O | Z | Z |
| | PF0 to PF15 | Z | | | I | Z | Z | I | I | I | I |

[Legend]

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High-impedance

K: Input pins become high-impedance, and output pins retain their state.

Notes:
1. Output pins become high-impedance when the HIZ bit in standby control register 6 (STBCR6) is set to 1.
2. Becomes output when the HIZMEM bit in the common control register (CMNCR) is set to 1.
3. Becomes input during a power-on reset. Pull-up to prevent erroneous operation. Pull-down with a resistance of at least 1 MΩ as required.
4. Pulled-up inside the LSI when there is no input.
5. SH7137 only.

# B. Processing of Unused Pins

**Table B.1 Processing of Unused Pins**

| Pin | Processing |
|---|---|
| NMI | Fixed high-level (pull-up) |
| $\overline{\text{WDTOVF}}$ | Open (If pull-down is necessary, use a resistor rated at 1 Ω or greater) |
| AVref | AVref = Avcc |
| AVcc, AVss | AVcc = Vcc, AVss = Vss |
| $\overline{\text{ASEMD0}}$ | Fixed high-level (pull-up) |
| PF0 to PF15 | Connect to AVcc or AVss via a resistor |
| Input-only pins other than the above | Fixed (pull-up/pull-down) |
| I/O pins other than the above | Fixed at input pin setting (pull-up/pull-down) or set to output and left open |
| Output-only pins | Open |

Notes:
1. For pull-up or pull-down, connect to Vcc or GND via a resistor.
2. When using the H-UDI, pin processing is according to the specifications of the emulator.

# C. Pin States of Bus Related Signals

## Table C.1 Pin States of Bus Related Signals (1)

| Pin Name | | On-chip ROM Space | On-chip RAM Space | On-chip Peripheral Module Space |
|---|---|---|---|---|
| $\overline{CS0}$, $\overline{CS1}$ | | H | H | H |
| $\overline{RD}$ | R | H | H | H |
| | W | — | H | H |
| $\overline{WRL}$ | R | H | H | H |
| | W | — | H | H |
| A19 to A0 | | Address* | Address* | Address* |
| D7 to D0 | | High-Z | High-Z | High-Z |

[Legend]
R: Read
W: Write
Note: * Value of external space address that was previously accessed

## Table C.1 Pin States of Bus Related Signals (2)

| Pin Name | | External Space (Normal Space) 8-bit Space |
|---|---|---|
| $\overline{CS0}$, $\overline{CS1}$ | | Enabled |
| $\overline{RD}$ | R | L |
| | W | H |
| $\overline{WRL}$ | R | H |
| | W | L |
| A19 to A0 | | Address |
| D7 to D0 | | Data |

[Legend]
R: Read
W: Write
Enabled: Chip select signals corresponding to accessed areas = Low.
The other chip select signals = High.

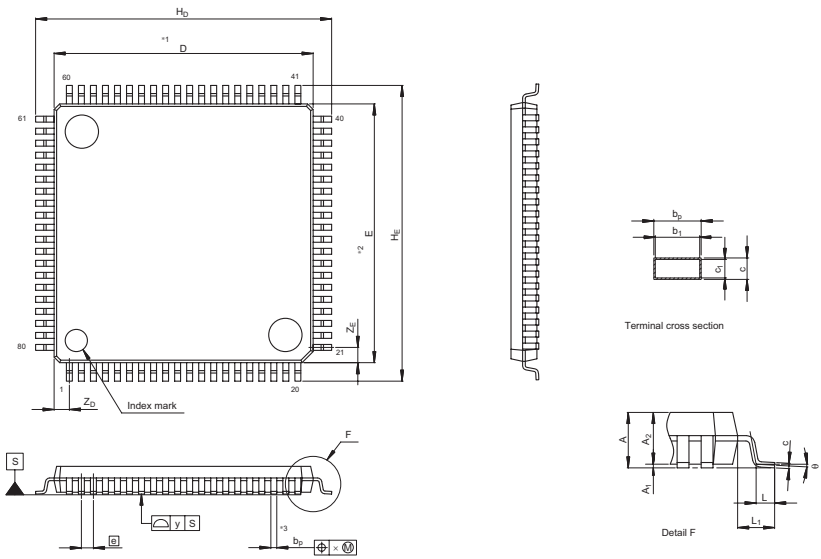
# D. Product Code Lineup

## Table D.1 Product Code Lineup

| Product Type | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Product Name** | **Classification** | **ROM Capacity** | **RAM Capacity** | **Application** | **Operating temperature** | **Part No.** | **Package (Package Code)** |
| SH7131 | F-ZTAT version | 128 Kbytes | 8 Kbytes | Industrial application | –40 to +85°C | R5F71313AD80FPV | LQFP1414-80 (FP-80WV) |
| | | 256 Kbytes | 16 Kbytes | | | R5F71314AD80FPV | |
| SH7132 | F-ZTAT version | 128 Kbytes | 8 Kbytes | Industrial application | –40 to +85°C | R5F71323AD80FPV | LQFP1414-100 (FP-100UV) |
| | | 256 Kbytes | 16 Kbytes | | | R5F71324AD80FPV | |
| SH7136 | F-ZTAT version | 256 Kbytes | 16 Kbytes | Consumer application | –20 to +85°C | R5F71364AN80FPV | LQFP1414-80 (FP-80WV) |
| | | | | Industrial application | –40 to +85°C | R5F71364AD80FPV | |
| SH7137 | F-ZTAT version | 256 Kbytes | 16 Kbytes | Consumer application | –20 to +85°C | R5F71374AN80FPV | LQFP1414-100 (FP-100UV) |
| | | | | Industrial application | –40 to +85°C | R5F71374AD80FPV | |

# E. Package Dimensions

| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LQFP80-14x14-0.65 | PLQP0080JA-A | FP-80W / FP-80WV | 0.6g |

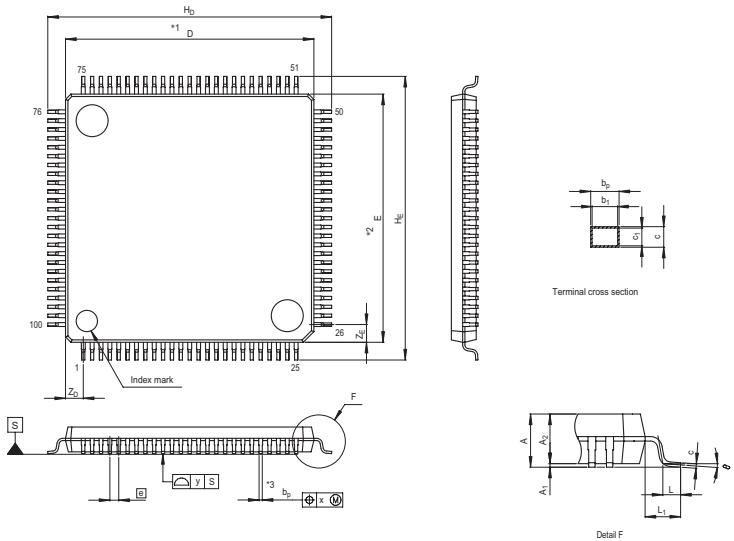



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 13.9 | 14.0 | 14.1 |
| E | 13.9 | 14.0 | 14.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 15.8 | 16.0 | 16.2 |
| $H_E$ | 15.8 | 16.0 | 16.2 |
| A | — | — | 1.7 |
| $A_1$ | 0.05 | 0.1 | 0.15 |
| $b_p$ | 0.27 | 0.32 | 0.37 |
| $b_1$ | — | 0.30 | — |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.65 | — |
| x | — | — | 0.13 |
| y | — | — | 0.10 |
| $Z_D$ | — | 0.825 | — |
| $Z_E$ | — | 0.825 | — |
| L | 0.35 | 0.5 | 0.65 |
| $L_1$ | — | 1.0 | — |

**Figure E.1 FP-80WV**

| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LQFP100-14x14-0.50 | PLQP0100KB-A | 100P6Q-A / FP-100U / FP-100UV | 0.6g |




NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 13.9 | 14.0 | 14.1 |
| E | 13.9 | 14.0 | 14.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 15.8 | 16.0 | 16.2 |
| $H_E$ | 15.8 | 16.0 | 16.2 |
| 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 | — | — | 0.08 |
| $Z_D$ | — | 1.0 | — |
| $Z_E$ | — | 1.0 | — |
| L | 0.35 | 0.5 | 0.65 |
| $L_1$ | — | 1.0 | — |

**Figure E.2 FP-100UV**

# Main Revisions for This Edition

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| All | — | SH7131 and SH7132 added |
| 1.1 Features of SH7131, SH7132, SH7136, and SH7137 | 1 | Description amended<br>This LSI also provides an external memory access support function to enable direct connection to various memory devices or peripheral LSIs (available only with the SH7132 and SH7137). |
| Table 1.1 Features of SH7131, SH7132, SH7136, and SH7137 | 2 | Table amended |

| Items | Specification |
|---|---|
| Operating modes | • Operating modes<br>Single chip mode<br>Extended ROM enabled mode (Only in SH7132/SH7137)<br>Extended ROM disabled mode (Only in SH7132/SH7137)<br>• Operating states<br>Program execution state<br>Exception handling state<br>Bus release state (Only in SH7132/SH7137)<br>• Power-down modes<br>Sleep mode<br>Software standby mode (Only in SH7136/SH7137)<br>Deep software standby mode (Only in SH7136/SH7137)<br>Module standby mode |
| User break controller (UBC)<br>(SH7132 and SH7137 only) | • Addresses, data values, type of access, and data size can all be set as break conditions<br>• Supports a sequential break function<br>• Two break channels |

3

| Items | Specification |
|---|---|
| On-chip ROM | • 128 Kbytes (Only in SH7131/SH7132) or 256 Kbytes |
| On-chip RAM | • 8 Kbytes (Only in SH7131/SH7132) or 16 Kbytes |
| Bus state controller (BSC) | • Address space: A maximum 1 Mbyte for each of two areas (CS0 and CS1) (Only in SH7132/SH7137)<br>• 8-bit external bus (Only in SH7132/SH7137)<br>• The following features settable for each area independently<br>Number of access wait cycles<br>Idle wait cycle insertion<br>Supports SRAM<br>• Outputs a chip select signal according to the target area |

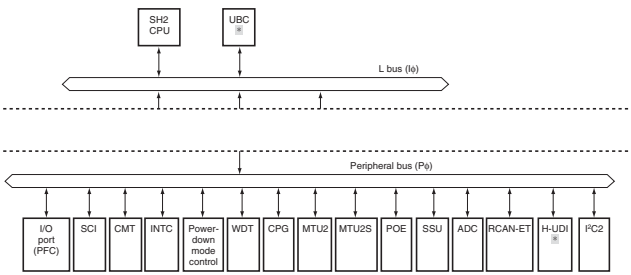
4

| Items | Specification |
|---|---|
| User debugging interface (H-UDI) (SH7136 and SH7137 only) | • Supports the E10A emulator |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 1.1 Features of SH7131, SH7132, SH7136, and SH7137<br>Table 1.1 Features of SH7131, SH7132, SH7136, and SH7137 | 6 | |

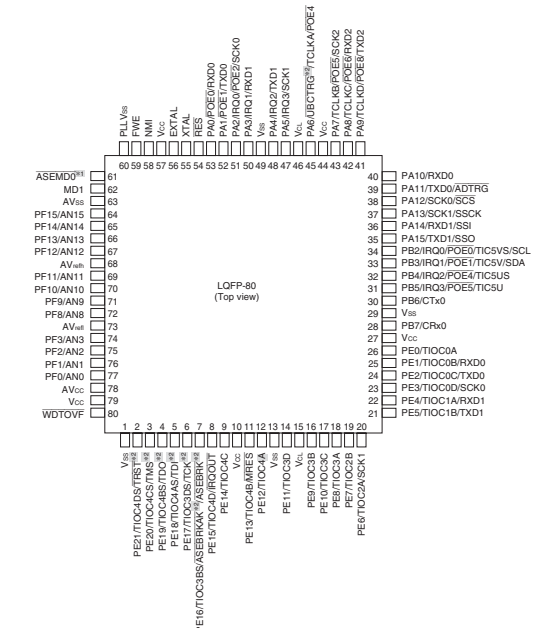
| Items | Specification |
|---|---|
| A/D converter (ADC) | • 12 bits × 16 channels (in SH7132/SH7137)<br>• 12 bits × 12 channels (in SH7131/SH7136)<br>• Conversion request by external triggers, MTU2, or MTU2S<br>• Two sample-and-hold function units (one unit consists of three sample-and-hold circuits) (three channels can be sampled simultaneously by an unit) |
| I/O ports | • 57 general input/output pins and 16 general input-only pins (SH7132/SH7137)<br>• 44 general input/output pins and 12 general input-only pins (SH7131/SH7136)<br>• Input or output can be selected for each bit |
| Package | • LQFP1414-100 (0.5 pitch) (SH7132/SH7137)<br>• LQFP1414-80 (0.65 pitch) (SH7131/SH7136) |

| | | |
|---|---|---|
| 1.2 Block Diagram<br>Figure 1.1 Block Diagram | 7 | Figure amended and note added |




Note: * SH7136 and SH7137 only.

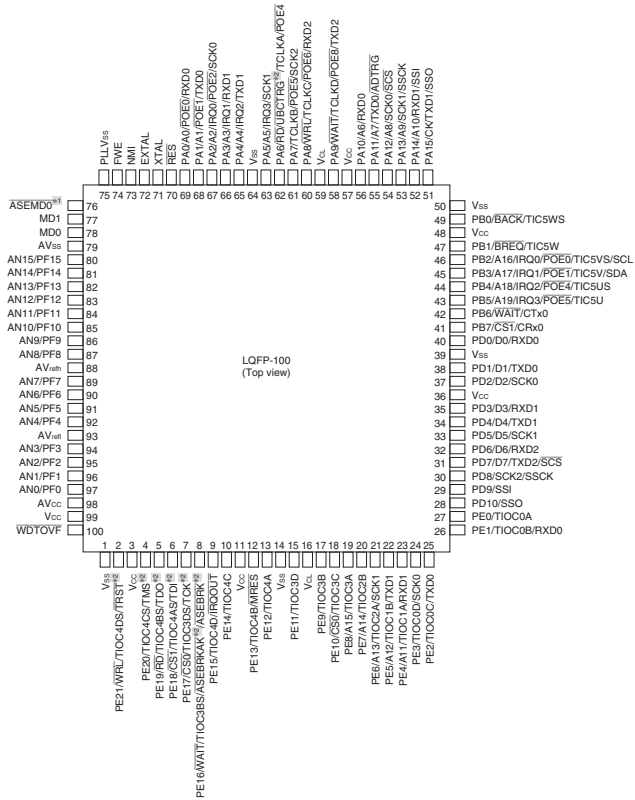
| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 1.2 Block Diagram<br>Figure 1.2 SH7131 and SH7136 Pin Assignments | 8 | Figure amended and note added |




Notes:

1. This pin is used by the E10A emulator. On the SH7131 it functions as a Vcc fixed pin and on the SH7136 as the ASEMD0 input pin.
2. Pin function enabled on the SH7136 only.

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 1.2 Block Diagram<br>Figure 1.3 SH7132 and SH7137 Pin Assignments | 9 | Figure amended and note added |




Notes:

1. This pin is used by the E10A emulator. On the SH7132 it functions as a Vcc fixed pin and on the SH7137 as the ASEMD0 input pin.
2. Pin function enabled on the SH7137 only.

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 1.4 Pin Functions<br>Table 1.2 Pin Functions | 15 | Table amended |

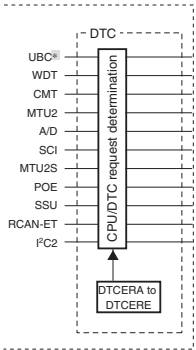
| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| User break controller (UBC) | $\overline{\text{UBCTRG}}$ | O | User break trigger output | Trigger output pin for UBC condition match<br>Available only in the SH7136/ SH7137. |
| User debugging interface (H-UDI) (SH7136 and SH7137 only) | TCK | I | Test clock | Test-clock input pin. |
| | TMS | I | Test mode select | Inputs the test-mode select signal. |
| | TDI | I | Test data input | Serial input pin for instructions and data. |
| | TDO | O | Test data output | Serial output pin for instructions and data. |
| | $\overline{\text{TRST}}$ | I | Test reset | Initialization-signal input pin. |

| | 16 | |
|---|---|---|

| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| E10A interface (SH7136 and SH7137 only) | $\overline{\text{ASEMDO}}$ | I | ASE mode | Sets the ASE mode. When this pin is driven low, the LSI enters ASE mode, and when driven high, the LSI operates in normal mode. Emulator dedicated functions can be used in the ASE mode. When nothing is input to this pin, it is pulled up internally. |
| | $\overline{\text{ASEBRK}}$ | I | Break request | E10A emulator break input |
| | $\overline{\text{ASEBRKAK}}$ | O | Break mode acknowledge | Indicates the E10A emulator has entered the break mode. |

| | | |
|---|---|---|
| 3.4 Address Map<br>Figure 3.1 Address Map for Each Operating Mode in SH7131 (128-Kbyte Flash Memory Version) | 52 | Figure added |
| Figure 3.2 Address Map for Each Operating Mode in SH7131 and SH7136 (256-Kbyte Flash Memory Version) | 53 | Figure title amended |
| Figure 3.3 Address Map for Each Operating Mode in SH7132 (128-Kbyte Flash Memory Version) | 54 | Figure added |
| Figure 3.4 Address Map for Each Operating Mode in SH7132 and SH7137 (256-Kbyte Flash Memory Version) | 55 | Figure title amended |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 4.1 Features<br>Table 4.1 Operating Clock for Each Module | 60 | Table amended and note added<br>**Operating Clock** / **Operating Module**<br>Internal clock (Iϕ) / CPU<br>UBC*<br>ROM<br>RAM<br>Note: * SH7136 and SH7137 only. |
| 5.1.1 Types of Exception Handling and Priority<br>Table 5.1 Types of Exceptions and Priority | 77 | Table amended and note amended<br>**Exception** / **Exceptio n Source**<br>Interrupt / User break (break before instruction execution)*3<br>Interrupt / User break (break after instruction execution or operand break)*3<br>Notes: 3. SH7136 and SH7137 only. |
| 5.1.3 Exception Handling Vector Table<br>Table 5.3 Vector Numbers and Vector Table Address Offsets | 79, 80 | Table amended and note amended<br>**Exception Handling Source** / **Vector Number** / **Vector Table Address Offset**<br>Interrupt / NMI / 11 / H'0000002C to H'0000002F<br>User break*1 / 12 / H'00000030 to H'00000033<br>**Exception Handling Source** / **Vector Number** / **Vector Table Address Offset**<br>On-chip peripheral module*2 / 72 / H'00000120 to H'00000123<br>: / :<br>255 / H'000003FC to H'000003FF<br>Notes:<br>1. SH7136 and SH7137 only.<br>2. For details on the vector numbers and vector table address offsets of on-chip peripheral module interrupts, see table 6.3 in section 6, Interrupt Controller (INTC). |
| 5.4.1 Interrupt Sources<br>Table 5.7 Interrupt Sources | 85 | Table amended and note added<br>**Type** / **Request Source** / **Number of Sources**<br>User break* / User break controller (UBC) / 1<br>Note: * SH7136 and SH7137 only. |
| 5.4.2 Interrupt Priority<br>Table 5.8 Interrupt Priority | 86 | Table amended and note added<br>**Type** / **Priority Level** / **Comment**<br>User break* / 15 / Fixed priority level. Can be masked.<br>Note: * SH7136 and SH7137 only. |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 6.1 Features<br>Figure 6.1 Block Diagram of INTC | 96 | Figure amended and note added<br>  <br><br>Note: * SH7136 and SH7137 only. |
| 6.4.3 User Break Interrupt (SH7136 and SH7137 only) | 109 | Title amended |
| 6.5 Interrupt Exception Handling Vector Table<br>Table 6.3 Interrupt Exception Handling Vectors and Priorities | 110 | Table amended and note amended<br>**Interrupt Source** / **Name** / **Vector No.** / **Vector Table Starting Address** / **IPR**<br>User break*1 / 12 / H'00000030 |
| | 113 | Notes:<br>1. SH7136 and SH7137 only.<br>2. Of the I²C2 interrupts, the vector address for the IINAKI interrupt is separated from others. |
| Section 7 User Break Controller (UBC) (SH7136 and SH7137 only) | 123 | Title amended |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 9.3.2 Address Map<br>Table 9.2 Address Map: SH7131/SH7132 (128-Kbyte Flash Memory Version) in Single-Chip Mode | 206 | Table title and table amended<br>Address / Area / Memory Type / Capacity / Bus Width<br>H'00000000 to H'0001FFFF / On-chip ROM / / 128 Kbytes / 32 bits<br>H'00020000 to H'FFFF9FFF / Reserved<br>H'FFFFA000 to H'FFFFBFFF / On-chip RAM / / 8 Kbytes / 32 bits |
| Table 9.3 Address Map: SH7131/SH7132/SH7136/SH7137 (256-Kbyte Flash Memory Version) in Single-Chip Mode | 206 | Table added |
| Table 9.4 Address Map: SH7132 (128-Kbyte Flash Memory Version) in On-Chip ROM Enabled Mode | 207 | Table added |
| Table 9.5 Address Map: SH7132 (128-Kbyte Flash Memory Version) in On-Chip ROM Disabled Mode) | 208 | Table added |
| Table 9.6 Address Map: SH7132/SH7137 (256-Kbyte Flash Memory Version) in On-Chip ROM-Enabled Mode | 209 | Table title amended |
| Table 9.7 Address Map: SH7132/SH7137 (256-Kbyte Flash Memory Version) in On-Chip ROM-Disabled Mode | 210 | Table title amended |

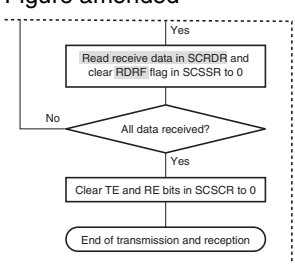
| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 10.3.32 Timer Waveform Control Register (TWCR) | 329 | Table amended |

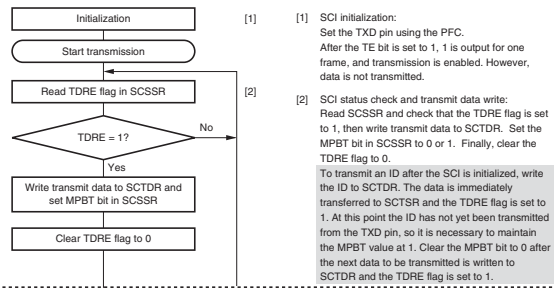
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | WRE | 0 | R/(W) | Initial Output Suppression Enable<br>Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.<br>The initial output is suppressed only when synchronous clearing 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.<br>For the Tb interval at the trough in complementary PWM mode, see figure 10.40.<br>0: Outputs the initial value specified in TOCR<br>1: Suppresses initial output<br>[Setting condition]<br>• When 1 is written to WRE after reading WRE = 0 |

| | | |
|---|---|---|
| 10.4.8 Complementary PWM Mode | 384 | Description amended |

14. …Even in the Tb interval at the trough, if synchronous clearing occurs in the initial value output period (indicated by (1) in figure 10.56) immediately after the counters start operation, initial value output is not suppressed.

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 10.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 10.4.8 Complementary PWM Mode<br>Figure 10.77 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0) | 403 | Figure replaced |
| Figure 10.78 Relationship between Bits T3AEN and T4VEN in Timer Interrupt Skipping Set Register (TITCR) and Buffer Transfer-Enabled Period | 404 | Figure replaced |
| 10.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode | 453, 454 | Newly added |
| 13.5 Interrupt Source | 532 | Newly added |
| 14.3.8 Serial Port Register (SCSPTR) | 550 | Table amended<br>Bit: 7 6 5 4 3 2 1 0<br>EIO - - - SPB1IO SPB1DT SPB0IO SPB0DT<br>Initial value: 0 0 0 0 0 - 0 1<br>R/W: R/W - - - R/W R/W R/W R/W |
| 14.4.3 Clock Synchronous Mode<br>Figure 14.16 Sample Flowchart for Transmitting/Receiving Serial Data | 583 | Figure amended<br>  |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 14.4.5 Multiprocessor Serial Data Transmission | 586 | Description amended<br>Figure 14.18 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SCSSR to 1 before transmission. Maintain the MPBT value at 1 until the ID transmission actually completes. |
| Figure 14.18 Sample Multiprocessor Serial Transmission Flowchart | | Figure amended<br>  <br> |
| 14.5 SCI Interrupt Sources and DTC | 590 | Description amended<br>When the ORER, FER, or PER flag in SCSSR is set to 1, an ERI interrupt request is generated. This request cannot be used to activate the DTC. It is possible to disable generation of RXI interrupt requests and allow only ERI interrupt requests to be generated during data reception processing. To accomplish this, set the RIE bit to 1 and the EIO bit in SCSPTR to 1. Note that setting the EIO bit to 1 will prevent the DTC from transferring received data because no ERI interrupt requests are generated. |
| 15.3.1 SS Control Register H (SSCRH) | 602 | Table amended |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1, 0 | CSS[1:0] | 01 | R/W | $\overline{\text{SCS}}$ Pin Select<br>Select that the $\overline{\text{SCS}}$ pin functions as $\overline{\text{SCS}}$ input or output.<br>00: Setting prohibited<br>01: Setting prohibited<br>10: Function as $\overline{\text{SCS}}$ automatic input/output (function as $\overline{\text{SCS}}$ input before and after transfer and output a low level during transfer)<br>11: Function as $\overline{\text{SCS}}$ automatic output (outputs a high level before and after transfer and outputs a low level during transfer) |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 15.3.2 SS Control Register L (SSCRL) | 603 | Table amended |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | FCLRM | 0 | R/W | Flag Clear Mode<br>Selects whether the SSRXI and SSTXI interrupt flags are cleared on writing to SSTDR or reading from SSRDR or on completion of DTC transfer. When using the DTC, set this bit to 0.<br>0: Flags are cleared when DTC transfer is completed (except when transfer counter value is H'0000)<br>1: Flags are cleared on SSTDR or SSRDR access |

| | | |
|---|---|---|
| 15.3.5 SS Status Register (SSSR) | 608 | Table amended |

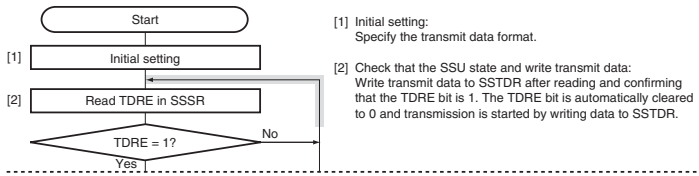
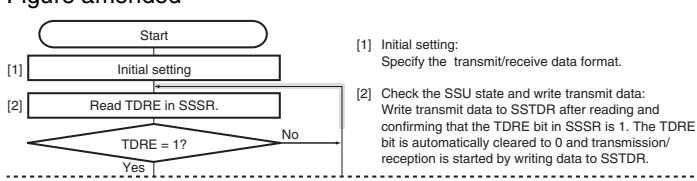
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | TDRE | 1 | R/W | Transmit Data Empty<br>Indicates whether or not SSTDR contains transmit data.<br>[Setting conditions]<br>• When the TE bit in SSER is 0<br>• When data is transferred from SSTDR to SSTRSR and SSTDR is ready to be written to.<br>[Clearing conditions]<br>• When writing 0 after reading TDRE = 1<br>• When writing data to SSTDR with TE = 1<br>• When the DTC is activated by an SSTXI interrupt and transmit data is written to SSTDR while the DISEL bit in MRB of the DTC is 0 (except when DTC transfer counter value is H'0000) |

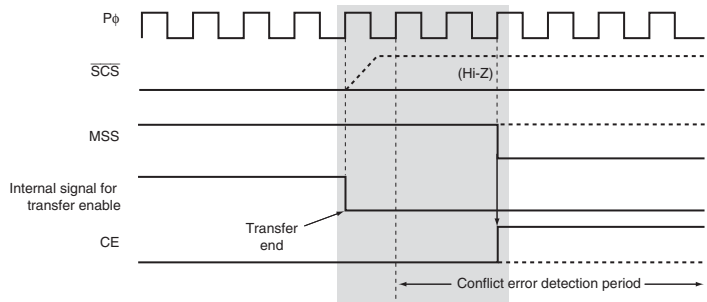
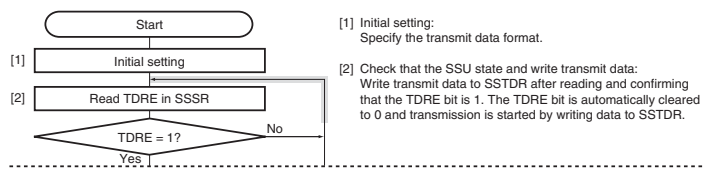
| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 15.3.5 SS Status Register (SSSR) | 609 | Table amended |

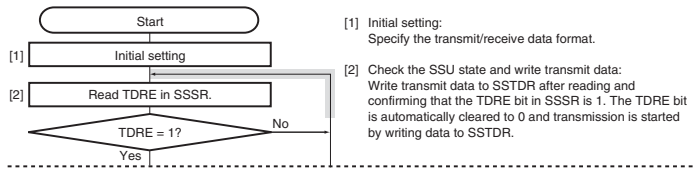
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | RDRF | 0 | R/W | Receive Data Register Full<br>Indicates whether or not SSRDR contains receive data.<br>[Setting condition]<br>• When receive data is transferred from SSTRSR to SSRDR after successful serial data reception<br>[Clearing conditions]<br>• When writing 0 after reading RDRF = 1<br>• When reading receive data from SSRDR<br>• When the DTC is activated by an SSRXI interrupt and receive data is read into SSRDR while the DISEL bit in MRB of the DTC is 0 (except when DTC transfer counter value is H'0000) |
| 0 | C E | 0 | R/W | Conflict/Incomplete Error<br>Indicates that a conflict error has occurred when 0 is externally input to the $\overline{SCS}$ pin with SSUMS = 0 (SSU mode) and MSS = 1 (master mode).<br>If the $\overline{SCS}$ pin level changes to 1 with SSUMS = 0 (SSU mode) and MSS = 0 (slave mode), an incomplete error occurs because it is determined that a master device has terminated the transfer. In addition, when SSUMS = 0 (SSU mode) and MSS = 0 (slave mode) and the next serial receive operation starts while RDRF = 1, an incomplete error occurs even if the data received from SSRDR is read before the completion of reception and RDRF is cleared to 0 before the $\overline{SCS}$ pin is set to 1. Data reception does not continue while the CE bit is set to 1. Serial transmission also does not continue. Reset the SSU internal sequencer by setting the SRES bit in SSCRL to 1 before resuming transfer after incomplete error.<br>[Setting conditions]<br>• When a low level is input to the $\overline{SCS}$ pin in master mode (the MSS bit in SSCRH is set to 1)<br>• When the $\overline{SCS}$ pin is changed to 1 during transfer in slave mode (the MSS bit in SSCRH is cleared to 0)<br>• When in slave mode (MSS = 0 in SSCRH), the next serial receive operation starts while RDRF = 1, and data is read from SSRDR before the completion of reception, after which the $\overline{SCS}$ pin is set to 1<br>[Clearing condition]<br>• When writing 0 after reading CE = 1 |

| | | |
|---|---|---|
| 15.4.4 Communication Modes and Pin Functions<br>Table 15.7 Communication Modes and Pin States of $\overline{SCS}$ Pin | 618 | Table amended |

| Communication Mode | Register Setting | | | | Pin State |
|---|---|---|---|---|---|
| | SSUMS | MSS | CSS1 | CSS0 | $\overline{SCS}$ |
| SSU communication mode | 0 | 0 | x | x | Input |
| | | 1 | 0 | 0 | — |
| | | | 0 | 1 | — |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 15.4.5 SSU Mode<br>(2) Data Transmission<br>Figure 15.6 Flowchart Example of Data Transmission (SSU Mode) | 623 | Figure amended<br>  <br> |
| (3) Data Reception | 624 | Description added<br>When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORER bit in SSSR is set to 1. This indicates that an overrun error (OEI) has occurred. At this time, data reception is stopped. While the ORER bit in SSSR is set to 1, reception is not performed. To resume the reception, clear the ORER bit to 0.<br>When setting the SSU to slave mode to perform continuous reception, read SSRDR before starting the next receive operation. If the next receive operation starts before SSRDR is read and RDRF is cleared to 0, and SSRDR is read before reception completes, CE in SSSR is set to 1 after the completion of reception.<br>In addition, if the next receive operation starts before SSRDR is read and RDRF is cleared to 0, and SSRDR is not read until after reception completes, the receive data is discarded even though neither CE nor ORER in SSSR is set to 1. |
| (4) Data Transmission/Reception | 627 | Description added<br>Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bit to 1.<br>If the value of RDRF is 1 when the 8th clock rises, ORER in SSSR is set to 1, an overrun error occurs, and reception halts. Receive operation is not possible while ORER is set to 1. To restart reception, first clear ORER to 0. |
| Figure 15.9 Flowchart Example of Simultaneous Transmission/Reception (SSU Mode) | | Figure amended<br>  <br> |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 15.4.6 $\overline{\text{SCS}}$ Pin Control and Conflict Error<br>Figure 15.11 Conflict Error Detection Timing (After Transfer End) | 629 | Figure amended<br>  <br> |
| 15.4.7 Clock Synchronous Communication Mode<br>(2) Data Transmission<br>Figure 15.14 Flowchart Example of Transmission Operation (Clock Synchronous Communication Mode) | 632 | Figure amended<br>  <br> |
| (3) Data Reception | 633 | Description amended<br>When 1-frame data has been received, the RDRF bit in SSSR is set to 1 and the receive data is stored in SSRDR. At this time, if the RIE bit is set to 1, an RXI interrupt is generated. The RDRF bit is automatically cleared to 0 by reading SSRDR.<br>When setting the SSU to slave mode to perform continuous reception, read SSRDR before starting the next receive operation. If the next receive operation starts before SSRDR is read and RDRF is cleared to 0, the integrity of subsequent data cannot be guaranteed. |
| (4) Data Transmission/Reception | 635 | Description added<br>Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bits to 1.<br>If the value of RDRF is 1 when the 8th clock rises, ORER in SSSR is set to 1, an overrun error occurs, and reception halts. Receive operation is not possible while ORER is set to 1. To restart reception, first clear ORER to 0. |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 15.4.7 Clock Synchronous Communication Mode<br>Figure 15.17 Flowchart Example of Simultaneous Transmission/Reception (Clock Synchronous Communication Mode) | 635 | Figure amended<br>  <br> |
| 15.6.4 Note for Reception Operations in SSU Slave Mode | 637 | Newly added |
| 15.6.5 Note on Master Transmission and Master Reception Operations in SSU Mode | 638 | Newly added |
| 15.6.6 Note on DTC Transfers | | Newly added |
| 16.3.2 $I^2C$ Bus Control Register 2 (ICCR2) | 646 | Table amended |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | BBSY | 0 | R/W | Bus Busy<br>This bit enables to confirm whether the $I^2C$ bus is occupied or released and to issue start/stop conditions in master mode. With the clock synchronous serial format, this bit is always read as 0. With the $I^2C$ 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. To issue a start condition, simultaneously write 1 to BBSY and 0 to SCP. Follow this procedure also when transmitting a repeated start condition. To issue a stop condition, simultaneously write 0 to BBSY and 0 to SCP. |
| 6 | SCP | 1 | R/W | Start/Stop Issue Condition Disable<br>The SCP bit controls the issue of start/stop conditions in master mode.<br>To issue a start condition, simultaneously write 1 to BBSY and 0 to SCP. A repeated start condition is issued in the same way. To issue a stop condition, simultaneously write 0 to BBSY and 0 to SCP. This bit is always read as 1. Even if 1 is written to this bit, the data will not be stored. |

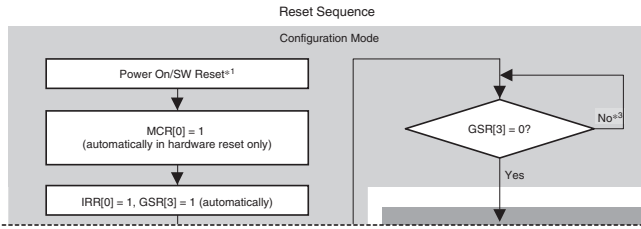
| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 16.3.4 I²C Bus Interrupt Enable Register (ICIER) | 650 | Table amended |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | RIE | 0 | R/W | Receive Interrupt Enable<br>RIE enables or disables the receive data full interrupt request (IIRXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. IIRXI can be canceled by clearing the RDRF or RIE bit to 0.<br>0: Receive data full interrupt request (IIRXI) are disabled.<br>1: Receive data full interrupt request (IIRXI) are enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | NAKIE | 0 | R/W | NACK Receive Interrupt Enable<br>NAKIE enables or disables the NACK detection/arbitration lost/overrun error interrupt request (IINAKI) when the NACKF or AL/OVE bit in ICSR is set to 1. IINAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0.<br>0: NACK receive interrupt request (IINAKI) is disabled.<br>1: NACK receive interrupt request (IINAKI) is enabled. |

| | | |
|---|---|---|
| 17.1 Features<br>Figure 17.1 Block Diagram of A/D Converter | 685 | Figure note amended<br>Note: Pins AN4 to AN7 are available only in the SH7132/SH7137. ADDR4 to ADDR7 registers are available only in the SH7132/SH7137. |
| 17.5 Interrupt Sources and DTC Transfer Requests<br>Table 17.7 Interrupt Sources | 706 | Newly added |
| 18.4.1 CMT Interrupt Sources and DTC Activation<br>Table 18.2 Interrupt Source | 717 | Newly added |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 19.4.2 Configuration of RCAN-ET | 768 | Figure and note amended |




Notes:

3. It takes about one bit time quantum for the value to reach 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.

Section 20 Pin Function Controller (PFC)

Table 20.1 SH7131/SH7136 Multiplexed Pins (Port A) — 785

Table amended and note added

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) |
|---|---|---|---|---|
| A | PA6 I/O (port) | $\overline{\text{UBCTRG}}$ output (UBC)* | TCLKA input (MTU2) | $\overline{\text{POE4}}$ input (POE) |

Note: * Function enabled on the SH7136 only.

Table 20.2 SH7132/SH7137 Multiplexed Pins (Port A) — 786

Table amended and note added

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) | Function 5 (Related Module) |
|---|---|---|---|---|---|
| A | PA6 I/O (port) | $\overline{\text{RD}}$ output (BSC) | $\overline{\text{UBCTRG}}$ output (UBC)* | TCLKA input (MTU2) | $\overline{\text{POE4}}$ input (POE) |

Note: * Function enabled on the SH7137 only.

Table 20.6 SH7131/SH7136 Multiplexed Pins (Port E) — 788

Table amended and note added

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) |
|---|---|---|---|---|
| E | PE16 I/O (port) | TIOC3BS I/O (MTU2S) | $\overline{\text{ASEBRKAK}}$ output (E10A)* | $\overline{\text{ASEBRK}}$ input (E10A)* |
| | PE17 I/O (port) | TIOC3DS I/O (MTU2S) | TCK input (H-UDI)* | |
| | PE18 I/O (port) | TIOC4AS I/O (MTU2S) | TDI input (H-UDI)* | |
| | PE19 I/O (port) | TIOC4BS I/O (MTU2S) | TDO output (H-UDI)* | |
| | PE20 I/O (port) | TIOC4CS I/O (MTU2S) | TMS input (H-UDI)* | |
| | PE21 I/O (port) | TIOC4DS I/O (MTU2S) | $\overline{\text{TRST}}$ input (H-UDI)* | |

Note: * Function enabled on the SH7136 only.

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 20 Pin Function Controller (PFC)<br>Table 20.7 SH7132/SH7137 Multiplexed Pins (Port E) | 789 | Table amended and note added |

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) | Function 5 (Related Module) |
|---|---|---|---|---|---|
| E | PE16 I/O (port) | $\overline{\text{WAIT}}$ input (BSC) | TIOC3BS I/O (MTU2S) | $\overline{\text{ASEBRKAK}}$ output (E10A)* | $\overline{\text{ASEBRK}}$ input (E10A)* |
| | PE17 I/O (port) | $\overline{\text{CS0}}$ output (BSC) | TIOC3DS I/O (MTU2S) | TCK input (H-UDI)* | |
| | PE18 I/O (port) | $\overline{\text{CS1}}$ output (BSC) | TIOC4AS I/O (MTU2S) | TDI input (H-UDI)* | |
| | PE19 I/O (port) | $\overline{\text{RD}}$ output (BSC) | TIOC4BS I/O (MTU2S) | TDO output (H-UDI)* | |
| | PE20 I/O (port) | TIOC4CS I/O (MTU2S) | TMS input (H-UDI)* | | |
| | PE21 I/O (port) | $\overline{\text{WRL}}$ output (BSC) | TIOC4DS I/O (MTU2S) | $\overline{\text{TRST}}$ input (H-UDI)* | |

Note: * Function enabled on the SH7137 only.

| | | |
|---|---|---|
| Table 20.10 SH7131/SH7136 Pin Functions in Each Operating Mode | 792 to 794 | Table amended and note amended |

| | Pin Name | |
|---|---|---|
| | Single-Chip Mode (MCU Mode 3) | |
| Pin No. | Initial Function | Functions Selectable by PFC |
| 59 | FWE*1 | FWE*1 |
| 61 | $\overline{\text{ASEMD0}}$*1 | $\overline{\text{ASEMD0}}$*1 |
| 45 | PA6 | PA6/$\overline{\text{UBCTRG}}$*2/TCLKA/POE4 |

| | Pin Name | |
|---|---|---|
| | Single-Chip Mode (MCU Mode 3) | |
| Pin No. | Initial Function | Functions Selectable by PFC |
| 7 | PE16/($\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$*1) | PE16/TIOC3BS |
| 6 | PE17/(TCK*1) | PE17/TIOC3DS |
| 5 | PE18/(TDI*1) | PE18/TIOC4AS |

| | Pin Name | |
|---|---|---|
| | Single-Chip Mode (MCU Mode 3) | |
| Pin No. | Initial Function | Functions Selectable by PFC |
| 4 | PE19/(TDO*1) PE1 | 9/TIOC4BS |
| 3 | PE20/(TMS*1) P | E20/TIOC4CS |
| 2 | PE21/($\overline{\text{TRST}}$*1) P | E21/TIOC4DS |

794

Notes:

1. Fixed to TMS, $\overline{\text{TRST}}$, TDI, TDO, TCK, and $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$ when using the E10A ($\overline{\text{ASEMD0}}$ = low).
2. Function enabled on the SH7136 only.

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 20 Pin Function Controller (PFC)<br>Table 20.11 SH7132/SH7137 Pin Functions in Each Operating Mode (1) | 795, 797 | Table amended and note amended |

| | Pin Name | |
|---|---|---|
| | On-Chip ROM Disabled (MCU Mode 0) | |
| Pin No. | Initial Function | Functions Selectable by PFC |
| 74 | FWE*1 | FWE*1 |
| 76 | $\overline{\text{ASEMD0}}$*1) | $\overline{\text{ASEMD0}}$*1) |
| 62 | $\overline{\text{RD}}$ | PA6/$\overline{\text{RD}}$/$\overline{\text{UBCTRG}}$*2/TCLKA/$\overline{\text{POE4}}$ |

| | Pin Name | |
|---|---|---|
| | On-Chip ROM Disabled (MCU Mode 0) | |
| Pin No. | Initial Function | Functions Selectable by PFC |
| 8 | PE16/($\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$*1) | PE16/$\overline{\text{WAIT}}$/TIOC3BS |
| 7 | PE17/(TCK*1) | PE17/$\overline{\text{CS0}}$/TIOC3DS |
| 6 | PE18/(TDI*1) | PE18/$\overline{\text{CS1}}$/TIOC4AS |
| 5 | PE19/(TDO*1) | PE19/$\overline{\text{RD}}$/TIOC4BS |
| 4 | PE20/(TMS*1) | PE20/TIOC4CS |
| 2 | PE21/($\overline{\text{TRST}}$*1) | PE21/$\overline{\text{WRL}}$/TIOC4DS |

798

Notes:

1. Fixed to TMS, $\overline{\text{TRST}}$, TDI, TDO, TCK, and $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$ when using the E10A ($\overline{\text{ASEMD0}}$ = low).
2. Function enabled on the SH7137 only.

| | | |
|---|---|---|
| Table 20.12 SH7132/SH7137 Pin Functions in Each Operating Mode (2) | 799, 801 | Table and note amended |

| | Pin Name | | | |
|---|---|---|---|---|
| | On-Chip ROM Enabled (MCU Mode 2) | | Single-Chip Mode (MCU Mode 3) | |
| Pin No. | Initial Function | Functions Selectable by PFC | Initial Function | Functions Selectable by PFC |
| 62 | PA6 | PA6/$\overline{\text{RD}}$/$\overline{\text{UBCTRG}}$*2/TCLKA/$\overline{\text{POE4}}$ | PA6 | PA6/$\overline{\text{UBCTRG}}$*2/TCLKA/$\overline{\text{POE4}}$ |

| | Pin Name | | | |
|---|---|---|---|---|
| | On-Chip ROM Enabled (MCU Mode 2) | | Single-Chip Mode (MCU Mode 3) | |
| Pin No. | Initial Function | Functions Selectable by PFC | Initial Function | Functions Selectable by PFC |
| 8 | PE16/($\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$*1) | PE16/$\overline{\text{WAIT}}$/TIOC3BS | PE16/($\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$*1) | PE16/TIOC3BS |
| 7 | PE17/(TCK*1) | PE17/$\overline{\text{CS0}}$/TIOC3DS | PE17/(TCK*1) | PE17/TIOC3DS |
| 6 | PE18/(TDI*1) | PE18/$\overline{\text{CS1}}$/TIOC4AS | PE18/(TDI*1) | PE18/TIOC4AS |
| 5 | PE19/(TDO*1) | PE19/$\overline{\text{RD}}$/TIOC4BS | PE19/(TDO*1) | PE19/TIOC4BS |
| 4 | PE20/(TMS*1) | PE20/TIOC4CS | PE20/(TMS*1) | PE20/TIOC4CS |
| 2 | PE21/($\overline{\text{TRST}}$*1) | PE21/$\overline{\text{WRL}}$/TIOC4DS | PE21/($\overline{\text{TRST}}$*1) | PE21/TIOC4DS |

802

Notes:

1. Fixed to TMS, $\overline{\text{TRST}}$, TDI, TDO, TCK, and $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$ when using the E10A ($\overline{\text{ASEMD0}}$ = low).
2. Function enabled on the SH7137 only.

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 20.1.2 Port A Control Registers L1 to L4 (PACRL1 to PACRL4)<br>SH7131/SH7136:<br>• Port A Control Register L2 (PACRL2) | 809 | Table and note added |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 | PA6MD2 | 0 | R/W | PA6 Mode |
| 9 | PA6MD1 | 0 | R/W | Select the function of the PA6/$\overline{\text{UBCTRG}}$/TCLKA/$\overline{\text{POE4}}$ pin. |
| 8 | PA6MD0 | 0 | R/W | 000: PA6 I/O (port)<br>001: TCLKA input (MTU2)<br>101: $\overline{\text{UBCTRG}}$ output (UBC)*<br>111: $\overline{\text{POE4}}$ input (POE)<br>Other than above: Setting prohibited |

Note: * Function enabled on the SH7136 only. Do not use this setting on the SH7131.

| | | |
|---|---|---|
| SH7132/SH7137:<br>• Port A Control Register L2 (PACRL2) | 815 | Table and note added |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 | PA6MD2 | 0 | R/W | PA6 Mode |
| 9 | PA6MD1 | 0*[1] | R/W | Select the function of the PA6/$\overline{\text{RD}}$/$\overline{\text{UBCTRG}}$/TCLKA/$\overline{\text{POE4}}$ pin. |
| 8 | PA6MD0 | 0*[1] | R/W | 000: PA6 I/O (port)<br>001: TCLKA input (MTU2)<br>011: $\overline{\text{RD}}$ output (BSC)*[2]<br>101: $\overline{\text{UBCTRG}}$ output (UBC)*[3]<br>111: $\overline{\text{POE4}}$ input (POE)<br>Other than above: Setting prohibited |

| | | |
|---|---|---|
| SH7132/SH7137:<br>• Port A Control Register L2 (PACRL2) | 816 | Notes:<br>1. The initial value is 1 in the on-chip ROM disabled external-extension mode.<br>2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.<br>3. Function enabled on the SH7137 only. Do not use this setting on the SH7132. |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 20.1.4 Port B Control Registers L1, L2 (PBCRL1, PBCRL2)<br>SH7131/SH7136:<br>• Port B Control Register L2 (PBCRL2) | 820, 821 | Table amended and note deleted |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 14 | PB7MD2 | 0 | R/W | PB7 Mode |
| 13 | PB7MD1 | 0 | R/W | Select the function of the PB7/CRx0 pin. |
| 12 | PB7MD0 | 0 | R/W | 000: PB7 I/O (port)<br>110: CRx0 input (RCAN-ET)<br>Other than above: Setting prohibited |
| 10 | PB6MD2 | 0 | R/W | PB6 Mode |
| 9 | PB6MD1 | 0 | R/W | Select the function of the PB6/CTx0 pin. |
| 8 | PB6MD0 | 0 | R/W | 000: PB6 I/O (port)<br>110: CTx0 output (RCAN-ET)<br>Other than above: Setting prohibited |
| 6 | PB5MD2 | 0 | R/W | PB5 Mode |
| 5 | PB5MD1 | 0 | R/W | Select the function of the PB5/IRQ3/$\overline{\text{POE5}}$/TIC5U pin. |
| 4 | PB5MD0 | 0 | R/W | 000: PB5 I/O (port)<br>001: IRQ3 input (INTC)<br>011: TIC5U input (MTU2)<br>111: $\overline{\text{POE5}}$ input (POE)<br>Other than above: Setting prohibited |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | PB4MD2 | 0 | R/W | PB4 Mode |
| 1 | PB4MD1 | 0 | R/W | Select the function of the PB4/IRQ2/$\overline{\text{POE4}}$/TIC5US pin. |
| 0 | PB4MD0 | 0 | R/W | 000: PB4 I/O (port)<br>001: IRQ2 input (INTC)<br>011: TIC5US input (MTU2S)<br>111: $\overline{\text{POE4}}$ input (POE)<br>Other than above: Setting prohibited |

| | | |
|---|---|---|
| SH7132/SH7137:<br>• Port B Control Register L2 (PBCRL2) | 823 | Table amended |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 14 | PB7MD2 | 0 | R/W | PB7 Mode |
| 13 | PB7MD1 | 0 | R/W | Select the function of the PB7/$\overline{\text{CS1}}$/CRx0 pin. |
| 12 | PB7MD0 | 0 | R/W | 000: PB7 I/O (port)<br>101: $\overline{\text{CS1}}$ output (BSC)*<br>110: CRx0 input (RCAN-ET)<br>Other than above: Setting prohibited |
| 10 | PB6MD2 | 0 | R/W | PB6 Mode |
| 9 | PB6MD1 | 0 | R/W | Select the function of the PB6/$\overline{\text{WAIT}}$/CTx0 pin. |
| 8 | PB6MD0 | 0 | R/W | 000: PB6 I/O (port)<br>101: $\overline{\text{WAIT}}$ input (BSC)*<br>110: CTx0 output (RCAN-ET)<br>Other than above: Setting prohibited |

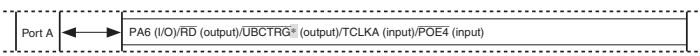
| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 20.1.4 Port B Control Registers L1, L2 (PBCRL1, PBCRL2)<br>SH7132/SH7137:<br>• Port B Control Register L1 (PBCRL1) | 825 | Table amended |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 14 | PB3MD2 | 0*1 | R/W | PB3 Mode |
| 13 | PB3MD1 | 0 | R/W | Select the function of the PB3/A17/IRQ1/$\overline{\text{POE1}}$/TIC5V/SDA pin. |
| 12 | PB3MD0 | 0*1 | R/W | 000: PB3 I/O (port)<br>001: IRQ1 input (INTC)<br>010: $\overline{\text{POE1}}$ input (POE)<br>011: TIC5V input (MTU2)<br>100: SDA I/O (IIC2)<br>101: A17 output (BSC)*2<br>Other than above: Setting prohibited |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 21.1 Port A<br>Figure 21.1 Port A (SH7131/SH7136) | 854 | Figure amended and note added |

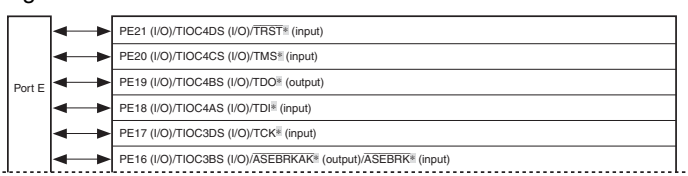
Note: * SH7136 only.

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Figure 21.2 Port A (SH7132/SH7137) | 855 | Figure amended and note added |



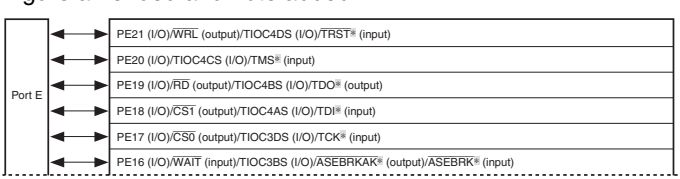

Note: * SH7137 only.

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 21.4 Port E<br>Figure 21.6 Port E (SH7131/SH7136) | 869 | Figure amended and note added |



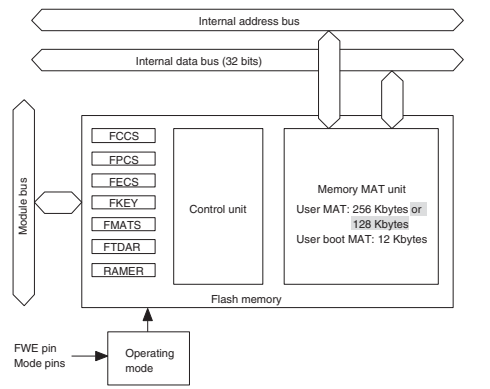
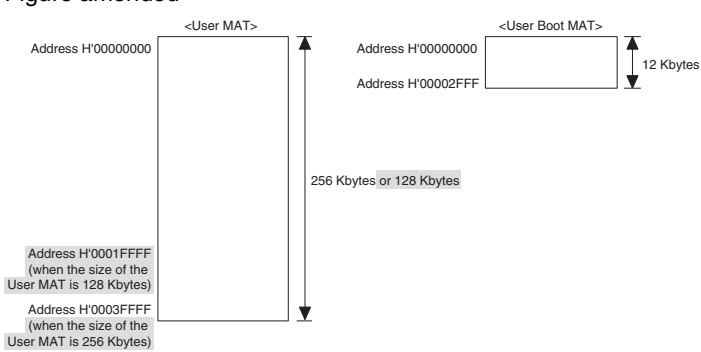

Note: * SH7136 only.

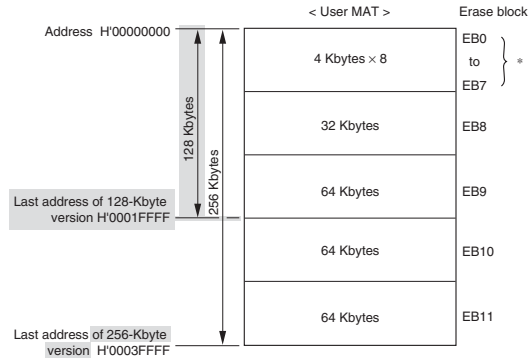
| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Figure 21.7 Port E (SH7132/SH7137) | 870 | Figure amended and note added |



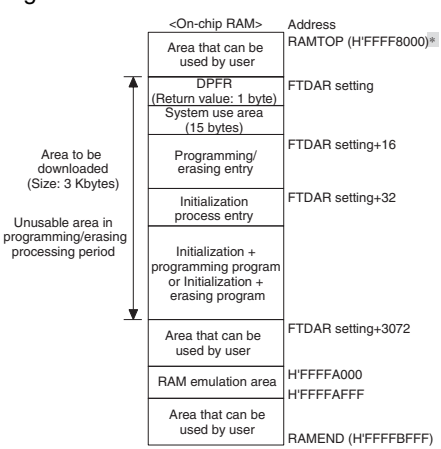

Note: * SH7137 only.

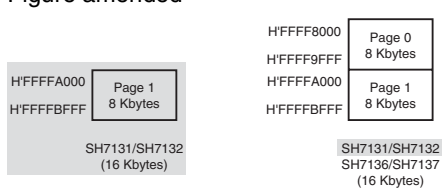
| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| Section 22 Flash Memory | 881 | Description amended<br>This LSI has 256-Kbyte or 128-Kbyte on-chip flash memory. The flash memory has the following features. |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 22.1 Features | 881 | Description amended<br>• Two flash-memory MATs, with one selected by the mode in which the LSI starts up …<br>— Size of the user MAT, from which booting-up proceeds after a power-on reset in user mode: 256 Kbytes or 128 Kbytes |
| 22.2.1 Block Diagram<br>Figure 22.1 Block Diagram of Flash Memory | 883 | Figure amended<br>  <br> |
| 22.2.4 Flash Memory Configuration | 887 | Description amended<br>This LSI's flash memory is configured by the 256-Kbyte or 128-Kbyte user MAT and 12-Kbyte user boot MAT. |
| Figure 22.3 Flash Memory Configuration | | Figure amended<br>  <br> |
| 22.2.5 Block Division | 888 | Description amended<br>The user MAT is divided into 64 Kbytes (256-Kbyte version: three blocks, 128-Kbyte version: one block), 32 Kbytes (one block), and 4 Kbytes (eight blocks) as shown in figure 22.4. |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 22.2.5 Block Division<br><br>Figure 22.4 Block Division of User MAT | 888 | Figure amended<br>  <br> |
| 22.4.3 Programming/Erasing Interface Parameters | 913 | Table amended |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | EBS[7:0] | Undefined | R/W | • 256-Kbyte flash memory<br>Set the erase-block number in the range from 0 to 11. 0 corresponds to the EB0 block and 11 corresponds to the EB11 block. An error occurs when a number other than 0 to 11 (H'00 to H'0B) is set.<br>• 128-Kbyte flash memory<br>Set the erase-block number in the range from 0 to 9. 0 corresponds to the EB0 block and 9 corresponds to the EB9 block. An error occurs when a number other than 0 to 9 (H'00 to H'09) is set. |

| | | |
|---|---|---|
| 22.5.2 User Program Mode<br><br>Figure 22.10 RAM Map after Download | 923 | Figure amended and note added<br>  <br><br>Note: * H'FFFFA000 on SH7131/SH7132. |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 22.7 Flash Memory Emulation in RAM<br>Figure 22.18 Example of Overlapped RAM Operation (256-Kbyte Flash Memory Version) | 941 | Figure title amended |
| Figure 22.19 Programming of Tuned Data (256-Kbyte Flash Memory Version) | 942 | Figure title amended |
| 22.8.3 Other Notes | 948 | Description added<br>5. Note on programming the product having a 128-Kbyte user MAT<br>If an attempt is made to program the product having a 128-Kbyte user MAT with more than 128 Kbytes, data programmed after the first 128 Kbytes are not guaranteed. |
| 22.9.1 Specifications of the Standard Serial Communications Interface in Boot Mode<br>(2) Device selection | 955 | Description amended<br>— Size (1 byte): Number of characters in the device code (fixed at 4) |
| 22.10 Programmer Mode | 986 | Description amended<br>Use a PROM programmer that supports the Renesas 128- or 256-Kbyte flash memory on-chip MCU device type (F-ZTATxxxx). |
| Section 23 RAM | 987 | Figure amended<br>  |
| 24.1.1 Types of Power-Down Modes | 989 | Description amended<br>• Sleep mode<br>• Software standby mode (SH7136 and SH7137 only)<br>• Deep software standby mode (SH7136 and SH7137 only) |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 24.1.1 Types of Power-Down Modes<br>Table 24.1 States of Power-Down Modes | 990 | Table amended and note added |

| Mode | Transition Method | State: CPG | State: CPU | State: CPU Register | State: On-Chip Memory | State: On-Chip Peripheral Modules | Canceling Procedure |
|---|---|---|---|---|---|---|---|
| Software standby* | Execute SLEEP instruction with STBY bit in STBCR1 and STBYMD bit in STBCR6 set to 1. | Halts | Halts | Held | Halts (contents retained) | Halt | • Interrupt by NMI or IRQ<br>• Power-on reset by the $\overline{\text{RES}}$ pin |
| Deep software standby* | Execute SLEEP instruction with STBY bit in STBCR1 set to 1 and STBYMD bit in STBCR6 cleared to 0. | Halts | Halts | Undefined | Halts (contents undefined) | Halt | • Power-on reset by the $\overline{\text{RES}}$ pin |

Notes: * SH7136 and SH7137 only.

| | | |
|---|---|---|
| 24.5 Software Standby Mode (SH7136 and SH7137 only) | 1001 | Title amended |
| 24.6 Deep Software Standby Mode (SH7136 and SH7137 only) | 1003 | Title amended |
| 24.8.2 Deep Software Standby Mode | — | Description deleted |
| 25.1 Register Address Table (In the Order of Addresses) | 1026 | Table amended |

| Register Name | Abbreviation | No. of Bits | Address | Module | Access Size | No. of Access Cycles | Connected Bus Width |
|---|---|---|---|---|---|---|---|
| Watchdog timer counter | WTCNT | 8 | H'FFFFE810 | WDT | 8*1, 16*2 | Pϕ (reference clock) | 16 bits |
| Watchdog timer control/status register | WTCSR | 8 | H'FFFFE812 | *1: Read<br>*2: Write | 8*1, 16*2 | B: 2*1<br>W: 2*2 | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 25.2 Register Bit List | 1046 | Table amended | | | | | | | |

| Register Abbreviation | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 | Module |
|---|---|---|---|---|---|---|---|---|---|
| IPRM | SSU | SSU | SSU | SSU | I²C2 | I²C2 | I²C2 | I²C2 | INTC |
| | RCAN-ET_0 | RCAN-ET_0 | RCAN-ET_0 | RCAN-ET_0 | — | — | — | — | |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 26.2 DC Characteristics<br>Table 26.2 DC Characteristics | 1065 | Table amended |

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Output high-level voltage | All output pins | $V_{OH}$ | $V_{CC}$–0.5 | — | — | V | $I_{OH}$ = –200 μA |
| | | | $V_{CC}$–1.0 | — | — | V | $I_{OH}$ = –1 mA |
| | TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS | | $V_{CC}$–1.0 | — | — | V | $I_{OH}$ = –5 mA |
| | PE9, PE11 to PE21 | | $V_{CC}$–2.0 | — | — | V | $I_{OH}$ = –5 mA |
| Output low-level voltage | All output pins | $V_{OL}$ | — | — | 0.4 | V | $I_{OL}$ = 1.6 mA |
| | SCL, SDA | | — | — | 0.4 | V | $I_{OL}$ = 3 mA |
| | | | — | — | 0.5 | V | $I_{OL}$ = 8 mA |
| | TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS | | — | — | 0.9 | V | $I_{OL}$ = 15 mA |
| | PE9, PE11 to PE21 | | — | — | 2.0 | V | $I_{OL}$ = 15 mA |

| | | |
|---|---|---|
| Table 26.3 DC Characteristics | 1068 | Table amended |

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Output high-level voltage | All output pins | $V_{OH}$ | $V_{CC}$–0.5 | — | — | V | $I_{OH}$ = –200 μA |
| | | | $V_{CC}$–1.0 | — | — | V | $I_{OH}$ = –1 mA |
| | TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS | | $V_{CC}$–1.0 | — | — | V | $I_{OH}$ = –5 mA |
| | PE9, PE11 to PE21 | | $V_{CC}$–2.0 | — | — | V | $I_{OH}$ = –5 mA |
| Output low-level voltage | All output pins | $V_{OL}$ | — | — | 0.4 | V | $I_{OL}$ = 1.6 mA |
| | SCL, SDA | | — | — | 0.4 | V | $I_{OL}$ = 3 mA |
| | | | — | — | 0.5 | V | $I_{OL}$ = 8 mA |
| | TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS | | — | — | 1.4 | V | $I_{OL}$ = 15 mA |
| | PE9, PE11 to PE21 | | — | — | 1.5 | V | $I_{OL}$ = 15 mA |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 26.5 Flash Memory Characteristics<br>Table 26.21 Flash Memory Characteristics | 1102 | Table amended |

| Item | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Programming time*1*2*4 | $t_P$ | — | 1 | 20 | ms/128 bytes |
| Erase time*1*2*4 | $t_E$ | — | 40 | 260 | ms/4 Kbyte block |
| | | — | 300 | 1500 | ms/32 Kbyte block |
| | | — | 600 | 3000 | ms/64 Kbyte block |
| Programming time (total)*1*2*4 | $\Sigma t_P$ | — | 2.3 | 12 | s/256 Kbytes |
| | | — | 1.1 | 6 | s/128 Kbytes |
| Erase time (total)*1*2*4 | $\Sigma t_E$ | — | 2.3 | 12 | s/256 Kbytes |
| | | — | 1.1 | 6 | s/128 Kbytes |
| Programming and erase time (total)*1*2*4 | $\Sigma t_{PE}$ | — | 4.6 | 24 | s/256 Kbytes |
| | | — | 2.2 | 12 | s/128 Kbytes |
| Reprogramming count | $N_{WEC}$ | 500*3 | — | — | Times |

| | | |
|---|---|---|
| A. Pin States<br>Table A.1 Pin States (SH7131/SH7136) | 1105, 1106 | Table and note amended |

| Pin Function | | Pin State | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Reset State | | Power-Down State | | | | |
| Type | Pin Name | Power-On | Manual | Deep Software Standby*4 | Software Standby*4 | Sleep | Oscillation Stop Detected | POE Function Used |
| UBC | $\overline{\text{UBCTRG}}$*4 | Z | O | Z | O*1 | O | O | O |

| | | |
|---|---|---|
| | 1107 | Notes: |
| | | 4. SH7136 only. |
| Table A.2 Pin States (SH7132/SH7137) | 1110 | Table and note amended |

| Pin Function | |
|---|---|
| Type | Pin Name |
| UBC | $\overline{\text{UBCTRG}}$*5 |

| | | |
|---|---|---|
| | 1111 | Notes: |
| | | 5. SH7137 only. |
| B. Processing of Unused Pins | 1111 | Newly added |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| D. Product Code Lineup<br>Table D.1 Product Code Lineup | 1113 | Table amended |

| Product Type | | | | | | | |
|---|---|---|---|---|---|---|---|
| Product Name | Classification | ROM Capacity | RAM Capacity | Application | Operating temperature | Part No. | Package (Package Code) |
| SH7131 | F-ZTAT version | 128 Kbytes | 8 Kbytes | Industrial application | –40 to +85˚C | R5F71313AD80FPV | LQFP1414-80 (FP-80WV) |
| | | 256 Kbytes | 16 Kbytes | | | R5F71314AD80FPV | |
| SH7132 | F-ZTAT version | 128 Kbytes | 8 Kbytes | Industrial application | –40 to +85˚C | R5F71323AD80FPV | LQFP1414-100 (FP-100UV) |
| | | 256 Kbytes | 16 Kbytes | | | R5F71324AD80FPV | |
| SH7136 | F-ZTAT version | 256 Kbytes | 16 Kbytes | Consumer application | –20 to +85˚C | R5F71364AN80FPV | LQFP1414-80 (FP-80WV) |
| | | | | Industrial application | –40 to +85˚C | R5F71364AD80FPV | |
| SH7137 | F-ZTAT version | 256 Kbytes | 16 Kbytes | Consumer application | –20 to +85˚C | R5F71374AN80FPV | LQFP1414-100 (FP-100UV) |
| | | | | Industrial application | –40 to +85˚C | R5F71374AD80FPV | |

| | | |
|---|---|---|
| E. Package Dimensions<br>Figure E.1 FP-80WV | 1114 | Figure replaced |
| Figure E.2 FP-100UV | 1115 | Figure replaced |

# Index


## A

A/D conversion time ..... 702
A/D converter (ADC) ..... 683
A/D converter activation..... 423
A/D converter activation by MTU2 and MTU2S ..... 703
A/D converter characteristics..... 1101
A/D converter start request delaying function..... 406
Absolute accuracy..... 707
Absolute maximum ratings..... 1063
AC bus timing..... 1078
AC characteristics..... 1071
AC characteristics measurement conditions..... 1100
Access in view of LSI internal bus master ..... 236
Access size and data alignment ..... 222
Access wait control..... 226
Address error ..... 83, 92, 988
Address map ..... 205
Address map for each mailbox ..... 729
Address map for the operating modes ..... 52
Addressing modes..... 26
Arithmetic operation instructions ..... 39
Asynchronous mode ..... 533, 565

## B

Bit synchronous circuit ..... 679
Block transfer mode..... 186
Boot mode..... 918
Branch instructions ..... 43
Break comparison conditions..... 123
Break detection and processing ..... 594
Break on data access cycle..... 148
Bus arbitration ..... 232
Bus clock (Bϕ) ..... 57
Bus release state ..... 47
Bus state controller (BSC) ..... 203

## C

Calculating exception handling vector table addresses..... 80
CAN interface ..... 727
Chain transfer..... 187
Changing frequency ..... 71
Clock (MIϕ) for the MTU2S module ..... 57
Clock (MPϕ) for the MTU2 module ..... 57
Clock frequency control circuit..... 59
Clock operating mode ..... 62
Clock pulse generator (CPG) ..... 57
Clock synchronous mode ..... 533, 575
Clock synchronous serial format ($I^2C2$)..... 668
Clock timing ..... 1072
CMT interrupt sources ..... 717
Compare match timer (CMT) ..... 711
Complementary PWM mode ..... 362
Conflict between NMI interrupt and DTC activation..... 202
Connecting crystal resonator..... 72
Continuous scan mode ..... 699
Control signal timing ..... 1075
Controller area network (RCAN-ET)..... 723
CPU..... 17
Crystal oscillator ..... 59
$\overline{\text{CS}}$n assert period extension ..... 228

## D

Data transfer controller (DTC)..... 159
Data transfer instructions ..... 37
DC characteristics ..... 1064
Dead time compensation..... 417
Deep software standby mode ..... 1003

Definitions of A/D conversion accuracy ..... 707
Divider ..... 59
DTC activation ..... 422
DTC activation by interrupt ..... 198
DTC activation sources ..... 172
DTC bus release timing ..... 194
DTC execution status ..... 192
DTC interface ..... 781
DTC vector address ..... 174

## E

Error protection ..... 938
Exception handling ..... 77
Exception handling state ..... 47
External clock input method ..... 73
External pulse width measurement ..... 416
External trigger input timing ..... 704

## F

Features of instructions ..... 23
Flash memory ..... 881
Flash memory characteristics ..... 1102
Flash memory configuration ..... 887
Flash memory emulation in RAM ..... 940
Flow of the user break operation ..... 146
Full-scale error ..... 707
Function for detecting oscillator stop ..... 74

## G

General illegal instructions ..... 88
General registers ..... 19
Global-base register (GBR) ..... 20

## H

Halt mode ..... 769
Hardware protection ..... 937

## I

I/O ports ..... 853
$I^2C$ bus format ..... 658
$I^2C$ bus interface 2 ($I^2C2$) ..... 639
ID Reorder ..... 738
Illegal slot instructions ..... 88
Immediate data formats ..... 23
Initial user branch processing time ..... 947
Initial values of control register ..... 21
Initial values of general register ..... 21
Initial values of system register ..... 21
Initiation intervals of user branch processing ..... 947
Input sampling and A/D conversion time ..... 701
Instruction formats ..... 29
Instruction set ..... 33
Interrupt controller (INTC) ..... 95
Interrupt exception handling vector table ..... 110
Interrupt priority ..... 86
Interrupt response time ..... 118
Interrupt sequence ..... 114
Interrupts ..... 85
IRQ interrupts ..... 108

## L

List of registers ..... 1007
Local acceptance filter mask (LAFM) ..... 735
Location of transfer information and DTC vector table ..... 172
Logic operation instructions ..... 41

## M

Mailbox ..... 726
Mailbox control ..... 726
Mailbox structure ..... 730
Manual reset ..... 82
MCU extension mode ..... 51
MCU operating modes ..... 49

Message control field ..... 731
Message data fields ..... 736
Message receive sequence ..... 776
Message transmission sequence ..... 773
Micro processor interface (MPI) ..... 726
Module standby mode ..... 1004
Module standby mode setting 201, 596, 637, ..... 681, 719, 988
MTU2 functions ..... 242
MTU2 interrupts ..... 421
MTU2 output pin initialization ..... 455
MTU2–MTU2S synchronous operation ..... 410
MTU2S functions ..... 488
Multi-function timer pulse unit 2 (MTU2) ..... 241
Multi-function timer pulse unit 2S (MTU2S) ..... 487
Multiply and accumulate registers (MACH and MACL) ..... 21
Multiprocessor communication function ..... 584

## N

NMI interrupt ..... 108
Noise filter ..... 672
Nonlinearity error ..... 707
Normal space interface ..... 223
Normal transfer mode ..... 183
Note on changing operating mode ..... 56
Note on crystal resonator ..... 75
Notes on board design ..... 75, 709
Notes on connecting $V_{CL}$ capacitor ..... 1103
Notes on noise countermeasures ..... 710
Notes on register access (WDT) ..... 529
Notes on slot illegal instruction exception handling ..... 93

## O

Offset error ..... 707
On-board programming mode ..... 918
On-chip peripheral module interrupts ..... 109
Operating clock for each module ..... 60

## P

Package dimensions ..... 1114
PC trace ..... 150
Peripheral clock (Pϕ) ..... 57
Pin function controller (PFC) ..... 785
Pin states of bus related signals ..... 1112
Pin states of this LSI in each processing state ..... 1105
Port output enable (POE) ..... 495
Power-down modes ..... 989
Power-down state ..... 47
Power-on reset ..... 81
Procedure register (PR) ..... 21
Product code lineup ..... 1113
Program counter (PC) ..... 21
Program execution state ..... 47
Programmer mode ..... 986

## Q

Quantization error ..... 707

## R

RAM ..... 987
RCAN-ET bit rate calculation ..... 748
RCAN-ET interrupt sources ..... 780
RCAN-ET memory map ..... 728
RCAN-ET reset sequence ..... 768
Reconfiguration of Mailbox ..... 778
Register
  ABACK0 ..... 762
  ADANSR_0 and ADANSR_1 ..... 694
  ADCR_0 and ADCR_1 ..... 688
  ADDR0 to ADDR15 ..... 695
  ADSR_0 and ADSR_1 ..... 691

ADSTRGR_0 and ADSTRGR_1 ..... 692
BAMRA ..... 127
BAMRB..... 133
BARA ..... 127
BARB ..... 132
BBRA ..... 128
BBRB ..... 136
BCR0, BCR1 ..... 745
BDMRA ..... 131
BDMRB..... 135
BDRA ..... 130
BDRB ..... 134
BETR..... 143
BRCR ..... 138
BRDR ..... 145
BRSR..... 144
BSCEHR..... 171, 218
CMCNT ..... 715
CMCOR..... 715
CMCSR ..... 713
CMNCR..... 211
CMSTR..... 713
CRA ..... 166
CRB ..... 167
CS0BCR and CS1BCR..... 213
CS0WCR and CS1WCR..... 216
DAR (DTC) ..... 165
DPFR ..... 902
DTCCR..... 169
DTCERA to DTCERE..... 168
DTCVBR ..... 171
FCCS ..... 894
FEBS..... 913
FECS..... 897
FKEY..... 898
FMATS..... 899
FMPAR..... 908
FMPDR..... 909
FPCS..... 897
FPEFEQ..... 904
FPFR ..... 907, 910, 914
FRQCR ..... 67
FTDAR ..... 900
FUBRA ..... 905
GSR..... 743
ICCR1 ..... 643
ICCR2 ..... 646
ICDRR ..... 656
ICDRS..... 656
ICDRT ..... 656
ICIER ..... 650
ICMR ..... 648
ICR0..... 99
ICSR ..... 652
ICSR1 ..... 500
ICSR2 ..... 504
ICSR3 ..... 509
IFCR ..... 851
IMR..... 755
IPRA, IPRD to IPRF and IPRH to
IPRL..... 105
IRQCR ..... 100
IRQSR..... 102
IRR..... 750
MBIMR0..... 764
MCR ..... 737
MRA ..... 162
MRB ..... 163
NF2CYC ..... 657
OCSR1 ..... 503
OCSR2 ..... 507
OSCCR ..... 70
PACRL1..... 804
PACRL2..... 804
PACRL3..... 804
PACRL4..... 804
PADRL ..... 856
PAIORL ..... 804
PAPRL ..... 858
PBCRL1 ..... 819

PBCRL2 ..... 819
PBDRL ..... 860
PBIORL ..... 819
PBPRL ..... 863
PDCRL1 ..... 828
PDCRL2 ..... 828
PDCRL3 ..... 828
PDCRL4 ..... 828
PDDRL ..... 866
PDIORL ..... 827
PDPRL ..... 867
PECRH1 ..... 834
PECRH2 ..... 834
PECRL1 ..... 834
PECRL2 ..... 834
PECRL3 ..... 834
PECRL4 ..... 834
PEDRH ..... 871
PEDRL ..... 871
PEIORH ..... 833
PEIORL ..... 833
PEPRH ..... 874
PEPRL ..... 874
PFDRL ..... 878
POECR1 ..... 512
POECR2 ..... 514
RAMCR ..... 999
RAMER ..... 916
REC ..... 755
RFPR0 ..... 763
RXPR0 ..... 762
SAR (DTC) ..... 165
SAR ($I^2C$2) ..... 655
SCBRR (SCI) ..... 553
SCRDR ..... 537
SCRSR (SCI) ..... 537
SCSCR (SCI) ..... 541
SCSDCR ..... 552
SCSMR (SCI) ..... 538
SCSPTR (SCI) ..... 550
SCSSR ..... 544
SCTDR ..... 538
SCTSR (SCI) ..... 537
SPOER ..... 511
SSCR2 ..... 610
SSCRH ..... 601
SSCRL ..... 603
SSER ..... 606
SSMR ..... 604
SSRDR0 to SSRDR3 ..... 612
SSSR ..... 607
SSTDR0 to SSTDR3 ..... 611
SSTRSR ..... 613
STBCR1 ..... 992
STBCR2 ..... 993
STBCR3 ..... 994
STBCR4 ..... 996
STBCR5 ..... 997
STBCR6 ..... 998
TADCOBRA_4 ..... 300
TADCOBRB_4 ..... 300
TADCORA_4 ..... 300
TADCORB_4 ..... 300
TADCR ..... 297
TBTER ..... 325
TBTM ..... 292
TCBR ..... 322
TCDR ..... 321
TCNT ..... 301
TCNTCMPCLR ..... 278
TCNTS ..... 320
TCR ..... 252
TCSYSTR ..... 306
TDDR ..... 321
TDER ..... 326
TEC ..... 755
TGCR ..... 318
TGR ..... 301
TICCR ..... 293
TIER ..... 279

TIOR ..... 259
TITCNT ..... 324
TITCR ..... 322
TMDR ..... 256
TOCR1 ..... 311
TOCR2 ..... 314
TOER ..... 310
TOLBR ..... 317
TRWER ..... 309
TSR ..... 284
TSTR ..... 302
TSYCR ..... 295
TSYR ..... 304
TWCR ..... 327
TXACK0 ..... 761
TXCR0 ..... 760
TXPR1, TXPR0 ..... 758
UMSR ..... 765
WTCNT ..... 526
WTCSR ..... 527
Register address table (in the order from lower addresses) ..... 1008
Register bit list ..... 1029
Register data format ..... 22
Register states in each operating mode ..... 1050
Repeat transfer mode ..... 184
Reset state ..... 47
Reset-synchronized PWM mode ..... 359
RISC-type ..... 23

## S

SCI interrupt sources ..... 590
SCSPTR and SCI pins ..... 591
Sending a break signal ..... 594
Sequential break ..... 149
Serial communication interface (SCI) ..... 533
Shift instructions ..... 42
Single chip mode ..... 51
Single-cycle scan mode ..... 697
Sleep mode ..... 769, 1000
Software protection ..... 938
Software standby mode ..... 1001
SSU Interrupt sources ..... 636
SSU mode ..... 619

Stack after interrupt exception handling ..... 117
Stack states after exception handling ends ..... 90
Status register (SR) ..... 19
Synchronous serial communication unit (SSU) ..... 597
System control instructions ..... 44

## T

Target pins and conditions for high-impedance control ..... 517
Test mode settings ..... 766
Time quanta is defined ..... 745
Transfer clock ..... 614
Transfer information read skip function ..... 182
Transfer information writeback skip function ..... 183
Trap instructions ..... 87

## U

User boot mode ..... 932
User break controller (UBC) ..... 123
User break interrupt ..... 109
User break on instruction fetch cycle ..... 147
User MAT ..... 888
User program mode ..... 922
Using interval timer mode ..... 531
Using watchdog timer mode ..... 530

## V

Vector numbers and vector table address offsets ..... 79

Vector-base register (VBR) ..... 20
Watchdog timer (WDT) ..... 523

## W

Wait between access cycles ..... 229

**Renesas 32-Bit RISC Microcomputer**
**Hardware Manual**
**SH7137 Group**


| | |
|---|---|
| Publication Date: | Rev.1.00, September 21, 2007 |
| | Rev.3.00, January 18, 2010 |
| Published by: | Sales Strategic Planning Div. |
| | Renesas Technology Corp. |
| Edited by: | Customer Support Department |
| | Global Strategic Communication Div. |
| | Renesas Solutions Corp. |


© 2010. Renesas Technology Corp., All rights reserved. Printed in Japan.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

## RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

**Renesas Technology America, Inc.**
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

**Renesas Technology Europe Limited**
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

**Renesas Technology (Shanghai) Co., Ltd.**
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

**Renesas Technology Hong Kong Ltd.**
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

**Renesas Technology Taiwan Co., Ltd.**
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

**Renesas Technology Singapore Pte. Ltd.**
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

**Renesas Technology Korea Co., Ltd.**
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

**Renesas Technology Malaysia Sdn. Bhd**
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

# SH7137 Group
# Hardware Manual

Renesas Electronics Corporation

REJ09B0402-0300