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.

# 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.

32

# SH7137 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family

SH7131
SH7132
SH7136
SH7137

Renesas Electronics
www.renesas.com

Rev.3.00

document, please confirm the latest product information with a Renesas sales office. Also, please pay re and careful attention to additional and different information to be disclosed by Renesas such as that disc 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 inform included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the informatio light of the total system before deciding about the applicability of such information to the intended applica Renesas makes no representations, warranties or guaranties regarding the suitability of its products for a 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 rec especially high quality and reliability such as safety systems, or equipment or systems for transportation traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea commun transmission. If you are considering the use of our products for such purposes, please contact a Renesa sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth abo
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed
   (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 elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Re Technology Corp., its affiliated companies and their officers, directors, and employees against any and a damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with res 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 malfunct 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 spec 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 inju injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design fo 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 produc 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 high. You should implement safety measures so that Renesas products may not be easily detached from 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 writte approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in t document, Renesas semiconductor products, or if you have any other inquiries.

induced in the vicinity of LSI, an associated shoot-through current flows internal
malfunctions may occur due to the false recognition of the pin state as an input
Unused pins should be handled as described under Handling of Unused Pins in
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 regis
   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, t
   states of pins are not guaranteed from the moment when power is supplied unti
   reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power
   reset function are not guaranteed from the moment when power is supplied unti
   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 functi
   not access these addresses; the correct operation of LSI is not guaranteed if the
   accessed.

4. Clock Signals

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

   — When the clock signal is generated with an external resonator (or from an exter
   oscillator) during a reset, ensure that the reset line is only released after full stal
   of the clock signal. Moreover, when switching to a clock signal produced with ar
   external resonator (or by an external oscillator) while program execution is in pr
   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 numbe
   confirm that the change will not lead to problems.

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

- CPU and System-Control Modules
- On-Chip Peripheral Modules

The configuration of the functional description of each module differs according to th 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. Eac includes notes in relation to the descriptions given, and usage notes are given, as required 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 vers This does not include all of the revised contents. For details, see the actual locations in th manual.

11. Index

Objective: This manual was written to explain the hardware functions and electrical characteristics of the SH7131, SH7132, SH7136, and SH7137 Group to th users.

Refer to the SH-1/SH-2/SH-DSP Software Manual for a detailed descripti 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 on the CPU, system control functions, peripheral functions and electrical characteris
- 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 entr register. The addresses, bits, and initial values of the registers are summarized in sec List of Registers.

  Examples: Register name: The following notation is used for cases when th similar function, e.g. serial communication interf implemented on more than one channel:
  XXX_N (XXX is the register name and N is the number)

  Bit order: The MSB is on the left and the LSB is on the righ

  Number notation: Binary is B'xxxx, hexadecimal is H'xxxx, decima

  Signal notation: An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

| Document Title | Document N |
|---|---|
| SuperH™ RISC engine C/C++ Compiler, Assembler,<br>Optimizing Linkage Editor Compiler Package V.9.00 User's Manual | REJ10B0152 |
| SuperH™ RISC engine High-performance Embedded Workshop 3<br>User's Manual | REJ10B0025 |
| SuperH RISC engine High-performance Embedded Workshop 3 Tutorial | REJ10B0023 |

Application note:

| Document Title | Document N |
|---|---|
| SuperH RISC engine C/C++ Compiler Package Application Note | REJ05B0463 |

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

2.2 Register Configuration
2.2.1 General Registers (Rn)
2.2.2 Control Registers
2.2.3 System Registers
2.2.4 Initial Values of Registers
2.3 Data Formats
2.3.1 Register Data Format
2.3.2 Memory Data Formats
2.3.3 Immediate Data Formats
2.4 Features of Instructions
2.4.1 RISC Type
2.4.2 Addressing Modes
2.4.3 Instruction Formats
2.5 Instruction Set
2.5.1 Instruction Set by Type
2.5.2 Data Transfer Instructions
2.5.3 Arithmetic Operation Instructions
2.5.4 Logic Operation Instructions
2.5.5 Shift Instructions
2.5.6 Branch Instructions
2.5.7 System Control Instructions
2.6 Processing States

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

4.4.2 Oscillation Stop Detection Control Register (OSCCR)
4.5 Changing Frequency
4.6 Oscillator
4.6.1 Connecting Crystal Resonator
4.6.2 External Clock Input Method
4.7 Function for Detecting Oscillator Stop
4.8 Usage Notes
4.8.1 Note on Crystal Resonator
4.8.2 Notes on Board Design

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

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

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

7.3.11 Break Control Register (BRCR) ..........
7.3.12 Execution Times Break Register (BETR)..........
7.3.13 Branch Source Register (BRSR)..........
7.3.14 Branch Destination Register (BRDR)..........
7.4 Operation ..........
7.4.1 Flow of the User Break Operation..........
7.4.2 User Break on Instruction Fetch Cycle..........
7.4.3 Break on Data Access Cycle..........
7.4.4 Sequential Break..........
7.4.5 Value of Saved Program Counter ..........
7.4.6 PC Trace ..........
7.4.7 Usage Examples..........
7.5 Usage Notes ..........

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

8.6 DTC Activation by Interrupt ....................................................................................

8.7 Examples of Use of the DTC ....................................................................................

8.7.1 Normal Transfer Mode ....................................................................................

8.7.2 Chain Transfer when Counter = 0 ....................................................................................

8.8 Interrupt Sources ....................................................................................

8.9 Usage Notes ....................................................................................

8.9.1 Module Standby Mode Setting ....................................................................................

8.9.2 On-Chip RAM ....................................................................................

8.9.3 DTCE Bit Setting ....................................................................................

8.9.4 Chain Transfer ....................................................................................

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

8.9.6 Access to DTC Registers through DTC ....................................................................................

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

8.9.8 Notes on SCI as DTC Activation Sources ....................................................................................

8.9.9 Clearing Interrupt Source Flag ....................................................................................

8.9.10 Conflict between NMI Interrupt and DTC Activation ....................................................................................

8.9.11 Operation When a DTC Activation Request is Cancelled While in Progre

Section 9 Bus State Controller (BSC) ....................................................................................

9.1 Features ....................................................................................

9.2 Input/Output Pins ....................................................................................

9.3 Area Overview ....................................................................................

9.3.1 Area Division ....................................................................................

9.3.2 Address Map ....................................................................................

9.4 Register Descriptions ....................................................................................

9.4.1 Common Control Register (CMNCR) ....................................................................................

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

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

9.5.10 Access to External Memory by CPU ........

Section 10 Multi-Function Timer Pulse Unit 2 (MTU2)........

10.1 Features........

10.2 Input/Output Pins........

10.3 Register Descriptions........

10.3.1 Timer Control Register (TCR)........

10.3.2 Timer Mode Register (TMDR)........

10.3.3 Timer I/O Control Register (TIOR)........

10.3.4 Timer Compare Match Clear Register (TCNTCMPCLR)........

10.3.5 Timer Interrupt Enable Register (TIER)........

10.3.6 Timer Status Register (TSR)........

10.3.7 Timer Buffer Operation Transfer Mode Register (TBTM)........

10.3.8 Timer Input Capture Control Register (TICCR)........

10.3.9 Timer Synchronous Clear Register (TSYCR)........

10.3.10 Timer A/D Converter Start Request Control Register (TADCR) ........

10.3.11 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4)........

10.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4) ........

10.3.13 Timer Counter (TCNT)........

10.3.14 Timer General Register (TGR) ........

10.3.15 Timer Start Register (TSTR) ........

10.3.16 Timer Synchronous Register (TSYR)........

10.3.17 Timer Counter Synchronous Start Register (TCSYSTR) ........

10.3.18 Timer Read/Write Enable Register (TRWER) ........

10.3.19 Timer Output Master Enable Register (TOER) ........

10.3.20 Timer Output Control Register 1 (TOCR1)........

10.3.21 Timer Output Control Register 2 (TOCR2)........

10.3.33 Bus Master Interface ..........................................................................................
10.4 Operation ..........................................................................................................
10.4.1 Basic Functions ..........................................................................................
10.4.2 Synchronous Operation ..........................................................................................
10.4.3 Buffer Operation ..........................................................................................
10.4.4 Cascaded Operation ..........................................................................................
10.4.5 PWM Modes ..........................................................................................
10.4.6 Phase Counting Mode ..........................................................................................
10.4.7 Reset-Synchronized PWM Mode..........................................................................................
10.4.8 Complementary PWM Mode ..........................................................................................
10.4.9 A/D Converter Start Request Delaying Function..........................................................................................
10.4.10 MTU2–MTU2S Synchronous Operation..........................................................................................
10.4.11 External Pulse Width Measurement..........................................................................................
10.4.12 Dead Time Compensation..........................................................................................
10.4.13 TCNT Capture at Crest and/or Trough in Complementary PWM Operatio
10.5 Interrupt Sources..........................................................................................
10.5.1 Interrupt Sources and Priorities..........................................................................................
10.5.2 DTC Activation..........................................................................................
10.5.3 A/D Converter Activation..........................................................................................
10.6 Operation Timing..........................................................................................
10.6.1 Input/Output Timing ..........................................................................................
10.6.2 Interrupt Signal Timing..........................................................................................
10.7 Usage Notes ..........................................................................................
10.7.1 Module Standby Mode Setting ..........................................................................................
10.7.2 Input Clock Restrictions ..........................................................................................
10.7.3 Caution on Period Setting ..........................................................................................
10.7.4 Contention between TCNT Write and Clear Operations..........................................................................................
10.7.5 Contention between TCNT Write and Increment Operations..........................................................................................
10.7.6 Contention between TGR Write and Compare Match ..........................................................................................

10.7.18 Contention between TCNT Write and Overflow/Underflow.......................

10.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode ..........................................................

10.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode ..............................................................................................

10.7.21 Interrupts in Module Standby Mode ..........................................................

10.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection...

10.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode ...................................................

10.8 MTU2 Output Pin Initialization...........................................................................

10.8.1 Operating Modes ..........................................................................................

10.8.2 Reset Start Operation ....................................................................................

10.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc. ........

10.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc. ........................................................................

## Section 11 Multi-Function Timer Pulse Unit 2S (MTU2S) .......................

11.1 Input/Output Pins.....................................................................................................

11.2 Register Descriptions ...............................................................................................

## Section 12 Port Output Enable (POE) .........................................................

12.1 Features....................................................................................................................

12.2 Input/Output Pins.....................................................................................................

12.3 Register Descriptions ...............................................................................................

12.3.1 Input Level Control/Status Register 1 (ICSR1) ..........................................

12.3.2 Output Level Control/Status Register 1 (OCSR1).......................................

12.3.3 Input Level Control/Status Register 2 (ICSR2) ..........................................

12.3.4 Output Level Control/Status Register 2 (OCSR2).......................................

12.3.5 Input Level Control/Status Register 3 (ICSR3) ..........................................

Section 13 Watchdog Timer (WDT)........................................

13.1 Features........................................

13.2 Input/Output Pin for WDT........................................

13.3 Register Descriptions........................................

13.3.1 Watchdog Timer Counter (WTCNT)........................................

13.3.2 Watchdog Timer Control/Status Register (WTCSR)........................................

13.3.3 Notes on Register Access........................................

13.4 Operation........................................

13.4.1 Revoking Software Standbys........................................

13.4.2 Using Watchdog Timer Mode........................................

13.4.3 Using Interval Timer Mode........................................

13.5 Interrupt Source........................................

13.6 Usage Note........................................

13.6.1 WTCNT Setting Value........................................

Section 14 Serial Communication Interface (SCI)........................................

14.1 Features........................................

14.2 Input/Output Pins........................................

14.3 Register Descriptions........................................

14.3.1 Receive Shift Register (SCRSR)........................................

14.3.2 Receive Data Register (SCRDR)........................................

14.3.3 Transmit Shift Register (SCTSR)........................................

14.3.4 Transmit Data Register (SCTDR)........................................

14.3.5 Serial Mode Register (SCSMR)........................................

14.3.6 Serial Control Register (SCSCR)........................................

14.3.7 Serial Status Register (SCSSR)........................................

14.3.8 Serial Port Register (SCSPTR)........................................

14.3.9 Serial Direction Control Register (SCSDCR)........................................

14.3.10 Bit Rate Register (SCBRR)........................................

14.7.2 Multiple Receive Error Occurrence
14.7.3 Break Detection and Processing
14.7.4 Sending a Break Signal
14.7.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode
14.7.6 Note on Using DTC
14.7.7 Note on Using External Clock in Clock Synchronous Mode
14.7.8 Module Standby Mode Setting

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

16.2 Input/Output Pins ..................
16.3 Register Descriptions ..................
16.3.1 I$^2$C Bus Control Register 1 (ICCR1) ..................
16.3.2 I$^2$C Bus Control Register 2 (ICCR2) ..................
16.3.3 I$^2$C Bus Mode Register (ICMR) ..................
16.3.4 I$^2$C Bus Interrupt Enable Register (ICIER) ..................
16.3.5 I$^2$C Bus Status Register (ICSR) ..................
16.3.6 I$^2$C Bus Slave Address Register (SAR) ..................
16.3.7 I$^2$C Bus Transmit Data Register (ICDRT) ..................
16.3.8 I$^2$C Bus Receive Data Register (ICDRR) ..................
16.3.9 I$^2$C Bus Shift Register (ICDRS) ..................
16.3.10 NF2CYC Register (NF2CYC) ..................
16.4 Operation ..................
16.4.1 I$^2$C Bus Format ..................
16.4.2 Master Transmit Operation ..................
16.4.3 Master Receive Operation ..................
16.4.4 Slave Transmit Operation ..................
16.4.5 Slave Receive Operation ..................
16.4.6 Clock Synchronous Serial Format ..................
16.4.7 Noise Filter ..................
16.4.8 Example of Use ..................
16.5 I$^2$C2 Interrupt Sources ..................
16.6 Operation Using the DTC ..................
16.7 Bit Synchronous Circuit ..................
16.8 Usage Note ..................
16.8.1 Module Standby Mode Setting ..................
16.8.2 Issuance of Stop Condition and Repeated Start Condition ..................
16.8.3 Issuance of a Start Condition and Stop Condition in Sequence ..................
16.8.4 Settings for Multi-Master Operation ..................

17.3.4 A/D Analog Input Channel Select Registers_0 and _1 (ADANSR_0 and ADANSR_1)
17.3.5 A/D Data Registers 0 to 15 (ADDR0 to ADDR15)
17.3.6 CPU Interface
17.4 Operation
17.4.1 Single-Cycle Scan Mode
17.4.2 Continuous Scan Mode
17.4.3 Input Sampling and A/D Conversion Time
17.4.4 A/D Converter Activation by MTU2 and MTU2S
17.4.5 External Trigger Input Timing
17.4.6 Example of ADDR Auto-Clear Function
17.5 Interrupt Sources and DTC Transfer Requests
17.6 Definitions of A/D Conversion Accuracy
17.7 Usage Notes
17.7.1 Analog Input Voltage Range
17.7.2 Relationship between AVcc, AVss and Vcc, Vss
17.7.3 Range of $\mathrm{AV}_{\mathrm{refh}}$ and $\mathrm{AV}_{\mathrm{refl}}$ Pin Settings
17.7.4 Notes on Board Design
17.7.5 Notes on Noise Countermeasures
17.7.6 Notes on Register Setting

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

18.5.5 Compare Match between CMCNT and CMCOR .........................................

Section 19 Controller Area Network (RCAN-ET) .................................

19.1 Summary ...........................................................................................

19.1.1 Overview ........................................................................................

19.1.2 Scope ..............................................................................................

19.1.3 Audience .........................................................................................

19.1.4 References .......................................................................................

19.1.5 Features ...........................................................................................

19.2 Architecture ......................................................................................

19.3 Programming Model – Overview ...................................................

19.3.1 Memory Map ..................................................................................

19.3.2 Mailbox Structure ...........................................................................

19.3.3 RCAN-ET Control Registers ..........................................................

19.3.4 RCAN-ET Mailbox Registers .........................................................

19.4 Application Note ...............................................................................

19.4.1 Test Mode Settings .........................................................................

19.4.2 Configuration of RCAN-ET ...........................................................

19.4.3 Message Transmission Sequence ...................................................

19.4.4 Message Receive Sequence ............................................................

19.4.5 Reconfiguration of Mailbox ...........................................................

19.5 Interrupt Sources ...............................................................................

19.6 DTC Interface ....................................................................................

19.7 CAN Bus Interface ............................................................................

19.8 Usage Notes .......................................................................................

19.8.1 Module Stop Mode .........................................................................

19.8.2 Reset ..............................................................................................

19.8.3 CAN Sleep Mode ...........................................................................

19.8.4 Register Access ..............................................................................

20.1.7 Port E I/O Registers L, H (PEIORL, PEIORH)

20.1.8 Port E Control Registers L1 to L4, H1, H2 (PECRL1 to PECRL4, PECRH1, PECRH2)

20.1.9 IRQOUT Function Control Register (IFCR)

20.2 Usage Notes

Section 21 I/O Ports

21.1 Port A

21.1.1 Register Descriptions

21.1.2 Port A Data Register L (PADRL)

21.1.3 Port A Port Register L (PAPRL)

21.2 Port B

21.2.1 Register Descriptions

21.2.2 Port B Data Register L (PBDRL)

21.2.3 Port B Port Register L (PBPRL)

21.3 Port D (SH7132/SH7137 Only)

21.3.1 Register Descriptions

21.3.2 Port D Data Register L (PDDRL)

21.3.3 Port D Port Register L (PDPRL)

21.4 Port E

21.4.1 Register Descriptions

21.4.2 Port E Data Registers H and L (PEDRH and PEDRL)

21.4.3 Port E Port Registers H and L (PEPRH and PEPRL)

21.5 Port F

21.5.1 Register Descriptions

21.5.2 Port F Data Register L (PFDRL)

Section 22 Flash Memory

22.1 Features

22.4.3 Programming/Erasing Interface Parameters ..........................................................
22.4.4 RAM Emulation Register (RAMER)..........................................................
22.5 On-Board Programming Mode ..........................................................
22.5.1 Boot Mode ..........................................................
22.5.2 User Program Mode..........................................................
22.5.3 User Boot Mode..........................................................
22.6 Protection..........................................................
22.6.1 Hardware Protection ..........................................................
22.6.2 Software Protection..........................................................
22.6.3 Error Protection..........................................................
22.7 Flash Memory Emulation in RAM ..........................................................
22.8 Usage Notes ..........................................................
22.8.1 Switching between User MAT and User Boot MAT..........................................................
22.8.2 Interrupts during Programming/Erasing ..........................................................
22.8.3 Other Notes ..........................................................
22.9 Supplementary Information ..........................................................
22.9.1 Specifications of the Standard Serial Communications Interface in Boot Mode ..........................................................
22.9.2 Areas for Storage of the Procedural Program and Data for Programming.
22.10 Programmer Mode ..........................................................

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

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

24.4.1 Transition to Sleep Mode........................................................................
24.4.2 Canceling Sleep Mode........................................................................
24.5 Software Standby Mode (SH7136 and SH7137 only)........................................
24.5.1 Transition to Software Standby Mode........................................................
24.5.2 Canceling Software Standby Mode.........................................................
24.6 Deep Software Standby Mode (SH7136 and SH7137 only)...................................
24.6.1 Transition to Deep Software Standby Mode...............................................
24.6.2 Canceling Deep Software Standby Mode...................................................
24.7 Module Standby Mode........................................................................
24.7.1 Transition to Module Standby Mode........................................................
24.7.2 Canceling Module Standby Function.......................................................
24.8 Usage Note........................................................................
24.8.1 Current Consumption while Waiting for Oscillation to be Stabilized .........
24.8.2 Executing the SLEEP Instruction ..........................................................

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

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

26.5 Flash Memory Characteristics ........................................................................

26.6 Usage Note........................................................................

26.6.1 Notes on Connecting $V_{CL}$ Capacitor........................................................................

Appendix ........................................................................

A. Pin States........................................................................

B. Processing of Unused Pins........................................................................

C. Pin States of Bus Related Signals ........................................................................

D. Product Code Lineup ........................................................................

E. Package Dimensions ........................................................................

Main Revisions for This Edition........................................................................

Index ........................................................................

the 32-bit internal-bus architecture enhances data processing power. With this CPU, it h
possible to assemble low-cost, high-performance, and high-functioning systems, even fo
applications that were previously impossible with microcomputers, such as real-time co
which demands high speeds.

In addition, this LSI includes on-chip peripheral functions necessary for system configu
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$^2$C bus interface 2 (I$^2$C2), and contr
network (RCAN-ET).

This LSI also provides an external memory access support function to enable direct con
various memory devices or peripheral LSIs (available only with the SH7132 and SH713

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

The version of on-chip ROM is F-ZTAT™ (Flexible Zero Turn Around Time)* that inc
memory. The flash memory can be programmed with a programmer that supports progra
this LSI, and can also be programmed and erased by software. This enables LSI chip to
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.

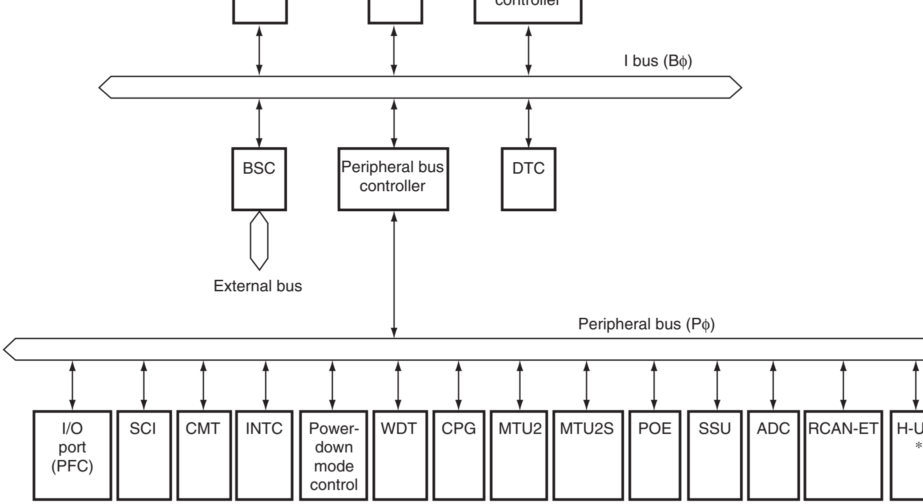
| | |
|---|---|
| | • On-chip multiplier: Multiplication operations (32 bits × 32 bits<br>executed in two to five cycles<br>• C language-oriented 62 basic instructions<br>Note: Some specifications on slot illegal instruction exception in this LSI differ from those of the conventional SH-2. Fo see section 5.8.4, Notes on Slot Illegal Instruction Excep 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 break conditions<br>• Supports a sequential break function<br>• Two break channels |

| | |
|---|---|
| | — 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 inter be done independently of the CPU transfer.<br>• Transfer mode selectable for each interrupt source (transfer 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 s<br>• Data transfer size can be specified as byte, word, or longwor<br>• The interrupt that activated the DTC can be issued to the CP<br>— A CPU interrupt can be requested after one data transfer completion.<br>— A CPU interrupt can be requested after all specified data completion. |
| Interrupt controller (INTC) | • Five external interrupt pins (NMI and IRQ3 to IRQ0)<br>• On-chip peripheral interrupts: Priority level set for each modu<br>• Vector addresses: A vector address for each interrupt source |

| | |
|---|---|
| | — 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 in 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 modes<br>• Synchronization of multiple counters<br>• Complementary PWM output mode<br>— Non-overlapping waveforms output for 6-phase inverter co<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 |

| | |
|---|---|
| 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, anc<br>• Full-duplex communication (transmission and reception exec simultaneously)<br>• Consecutive serial communication<br>• One channel |
| I$^2$C bus interface (IIC2) | • Conforming to Philips I$^2$C bus interface specifications<br>• Master mode and slave mode supported<br>• Continuous transmission/reception<br>• I$^2$C 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 bu reception only<br>• One channel |

| | |
|---|---|
| | (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 |

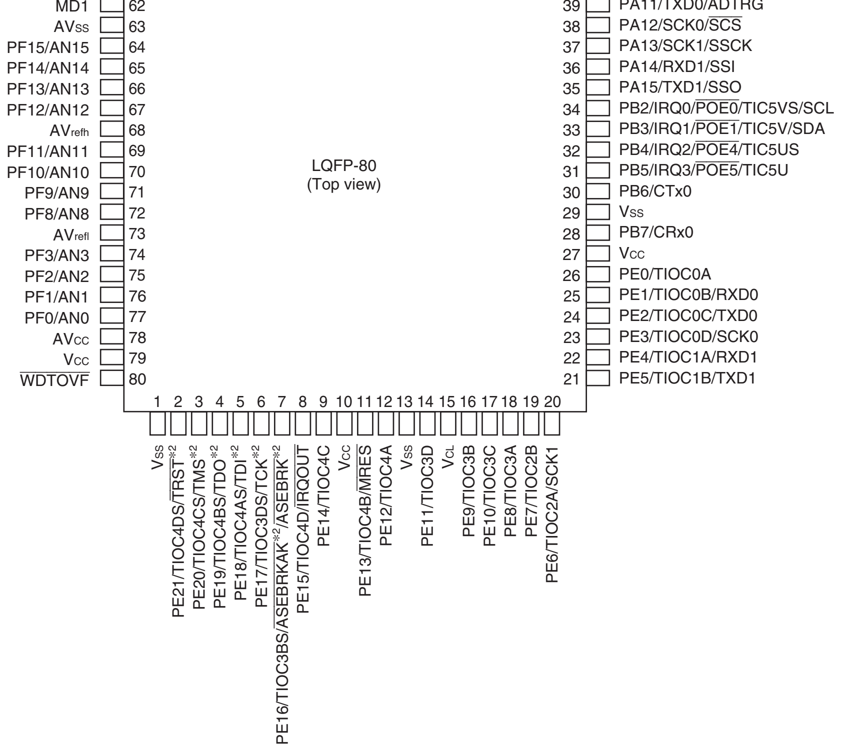


[Legend]

| | | | |
|---|---|---|---|
| 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$^2$C2: | I$^2$C Bus interface 2 |
| | | H-UDI: | User debugging interface |

Note: * SH7136 and SH7137 only.

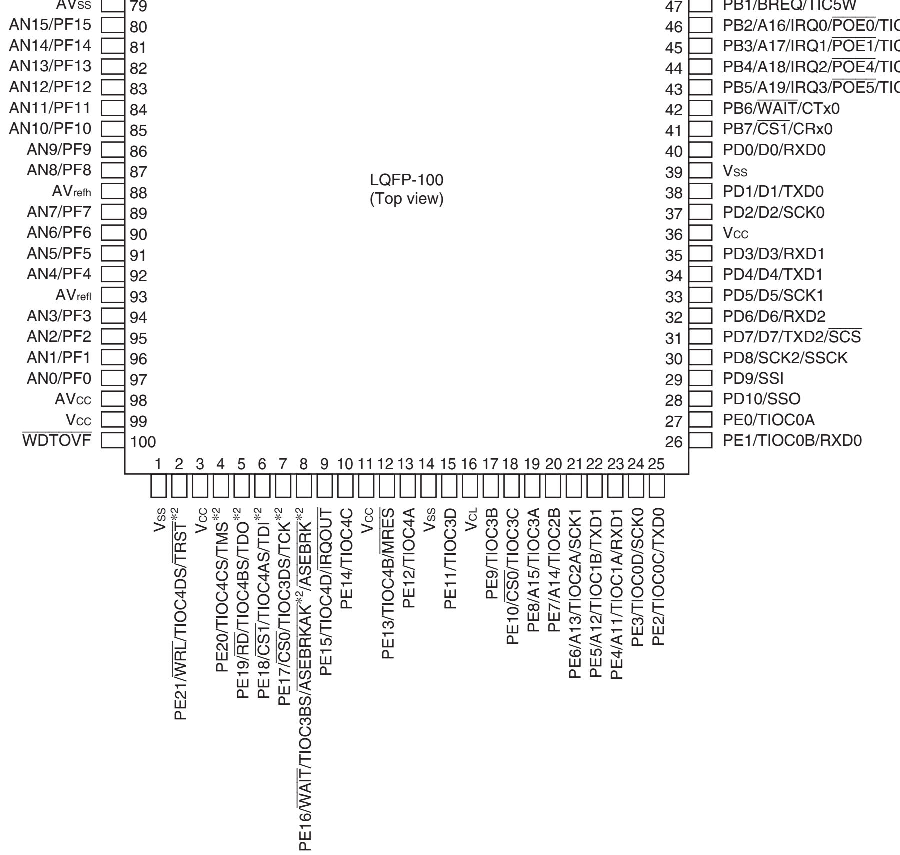
**Figure 1.1 Block Diagram**



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 SH713
as the ASEMD0 input pin.

2. Pin function enabled on the SH7137 only.

**Figure 1.3 SH7132 and SH7137 Pin Assignments**

| | | | | |
|---|---|---|---|---|
| | Vss | I | Ground | Ground pin<br>Connect all Vss pins to the s<br>power supply (0V). The LSI<br>operate if any pins are oper |
| | VCL | O | Internal step-down power supply | External capacitance pins fo<br>step-down power supply<br>Connect these pins to Vss v<br>μF capacitor (should be plac<br>to the pins). |
| Clock | PLLVss | I | PLL ground | Ground pin for the on-chip F<br>oscillator |
| | EXTAL | I | External clock | Connected to a crystal reso<br>An external clock signal ma<br>input to the EXTAL pin. |
| | XTAL | O | Crystal | Connected to a crystal reso |
| | CK | O | System clock | Supplies the system clock to<br>devices. The SH7131/SH71<br>not have this pin. |

|  |  |  |  |
|---|---|---|---|
| $\overline{\text{MRES}}$ | I | Manual reset | When low, this LSI enters t<br>manual reset state. |
| $\overline{\text{WDTOVF}}$ | O | Watchdog timer overflow | Output signal for the watch<br>overflow<br>Use a resistor of 1 MΩ or m<br>this pin needs to be pulled |
| $\overline{\text{BREQ}}$ | I | Bus-mastership request | Low when an external dev<br>requests the release of the<br>mastership. This pin is ava<br>on the SH7132/SH7137. |
| $\overline{\text{BACK}}$ | O | Bus-mastership request acknowledge | Indicates that the bus mas<br>has been released to an ex<br>device. Reception of the $\overline{\text{B}}$<br>signal informs the device w<br>output the $\overline{\text{BREQ}}$ signal tha<br>acquired the bus. This pin<br>available only on the<br>SH7132/SH7137. |

| | | | | |
|---|---|---|---|---|
| | | | | recognized even in the bus state. |
| Address bus | A19 to A0 | O | Address bus | Outputs addresses. This pir available only on the SH7132/SH7137. |
| Data bus | D7 to D0 | I/O | Data bus | 8-bit bidirectional bus. This available only on the SH7132/SH7137. |
| Bus control | $\overline{\text{CS1}}$, $\overline{\text{CS0}}$ | O | Chip select 1 and 0 | Chip-select signal for exterr memory or devices. This pir available only on the SH7132/SH7137. |
| | $\overline{\text{RD}}$ | O | Read | Indicates reading of data fro external devices. This pin is available only on the SH7132/SH7137. |
| | $\overline{\text{WRL}}$ | O | Write | Indicates a write access to l of the external data. This pir available only on the SH7132/SH7137. |
| | $\overline{\text{WAIT}}$ | I | Wait | Input signal for inserting a w into the bus cycles during a the external space. This pin available only on the SH7132/SH7137. |

| | | | | |
|---|---|---|---|---|
| | | | compare<br>(channel 1) | output/PWM output pins |
| | TIOC2A,<br>TIOC2B | I/O | MTU2 input capture/output compare (channel 2) | The TGRA_2 to TGRB_2 i capture input/output compa output/PWM output pins |
| | TIOC3A,<br>TIOC3B,<br>TIOC3C,<br>TIOC3D | I/O | MTU2 input capture/output compare (channel 3) | The TGRA_3 to TGRD_3 i capture input/output compa output/PWM output pins |
| | TIOC4A,<br>TIOC4B,<br>TIOC4C,<br>TIOC4D | I/O | MTU2 input capture/output compare (channel 4) | The TGRA_4 to TGRD_4 i capture input/output compa output/PWM output pins |
| | TIC5U,<br>TIC5V,<br>TIC5W | I | MTU2 input capture (channel 5) | The TGRU_5, TGRV_5, ar TGRW_5 input capture inp (The TIC5W pin is availabl the SH7132/SH7137) |
| Multi function timer-pulse unit 2S (MTU2S) | TIOC3BS,<br>TIOC3DS | I/O | MTU2S input capture/output compare (channel 3) | The TGRB_3S and TGRD_ capture input/output compa output/PWM output pins |
| | TIOC4AS,<br>TIOC4BS,<br>TIOC4CS,<br>TIOC4DS | I/O | MTU2S input capture/output compare (channel 4) | The TGRA_4S to TGRD_4 capture input/output compa output/PWM output pins |
| | TIC5US,<br>TIC5VS,<br>TIC5WS | I | MTU2S input capture (channel 5) | The TGRU_5S, TGRV_5S TGRW_5S input capture ir (The TIC5WS pin is availa the SH7132/SH7137) |

| | | | | |
|---|---|---|---|---|
| | SCK0 | | | |
| 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 b |
| | CRx0 | I | Receive data | Receive data pin for CAN bu |
| I$^2$C bus interface 2 (I$^2$C2) | SCL | I/O | I$^2$C clock input/output | I$^2$C bus clock input/output pi |
| | SDA | I/O | I$^2$C data input/output | I$^2$C bus data input/output pir |
| A/D converter (ADC) | AN15 to AN0 | I | Analog input pins | Analog input pins (AN15 to AN3 to AN0 for the SH7131/SH7136) |
| | $\overline{\text{ADTRG}}$ | I | A/D conversion trigger input | External trigger input pin for A/D conversion |
| | AVcc | I | Analog power supply | Power supply pin for the A/D converter<br>Connect it to the system pov supply (Vcc) when the A/D c is not used.<br>Connect all AVcc pins to the power supply (Vcc) The A/D converter does not work if a open. |

| | | | | |
|---|---|---|---|---|
| | AVrefl | I | Analog reference power supply (low) | Analog reference power su |
| I/O ports | PA15 to PA0 | I/O | General port | 16 bits of general input/out pins |
| | PB7 to PB0 | I/O | General port | 8 bits of general input/outp pins (PB7 to PB2 in the SH7131/SH7136) |
| | PD10 to PD0 | I/O | General port | 11 bits of general input/out pins (The SH7131/SH7136 have this pin) |
| | PE21 to PE0 | I/O | General port | 22 bits of general input/out pins |
| | PF15 to PF0 | I | General port | 16 bits of general input por (PF15 to PF8, PF3 to PF0 SH7131/SH7136) |
| User break controller (UBC) | $\overline{\text{UBCTRG}}$ | O | User break trigger output | Trigger output pin for UBC match<br>Available only in the SH71 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 selec |
| | TDI | I | Test data input | Serial input pin for instructi data. |
| | TDO | O | Test data output | Serial output pin for instruc data. |
| | $\overline{\text{TRST}}$ | I | Test reset | Initialization-signal input pi |

|  |  |  |  |
|---|---|---|---|
| ASEBRKAK | O | Break mode acknowledge | Indicates the E10A emulato<br>entered the break mode. |

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

Post-increment register indirect (@Rn+)

Pre-decrement register indirect (@-Rn)

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

Index register indirect (@R0, Rn)

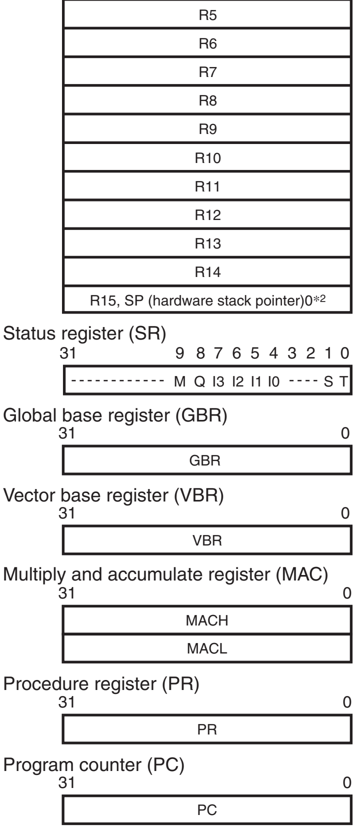
GBR indirect with displacement (@disp:8, GBR)

Index GBR indirect (@R0, GBR)

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

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

Immediate (#imm:8)


R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15, SP (hardware stack pointer)0*2
Status register (SR)
31
9 8 7 6 5 4 3 2 1 0
- - - - - - - - - - - - M Q I3 I2 I1 I0 - - - - S T
Global base register (GBR)
31
0
GBR
Vector base register (VBR)
31
0
VBR
Multiply and accumulate register (MAC)
31
0
MACH
MACL
Procedure register (PR)
31
0
PR
Program counter (PC)
31
0
PC

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**

(GBR), and vector base register (VBR). SR indicates a processing state. GBR is used as address in GBR indirect addressing mode for data transfer of on-chip peripheral module VBR is used as a base address of the exception handling (including interrupts) vector tal

- Status register (SR)

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

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | - | - | - | - | - | - | M | Q | I[3:0] |  |  |  | - | - |
| 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 |

| Bit | Bit name | Default | Read/ Write | Description |
|---|---|---|---|---|
| 31 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write v should always be 0. |
| 9 | M | Undefined | R/W | Used by the DIV0U, DIV0S, and DIV1 instru |
| 8 | Q | Undefined | R/W | Used by the DIV0U, DIV0S, and DIV1 instru |
| 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 v should always be 0. |

- Global-base register (GBR)

  This register indicates a base address in GBR indirect addressing mode. The GBR ind addressing mode is used for data transfer of the on-chip peripheral module registers ar operations.

- Vector-base register (VBR)

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

The PC indicates the point which is four bytes (two instructions) after the current exe
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 vect |
| 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 vect |

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 ac
from any address. Locate, however, word data at an address 2n, longword data at 4n. Othe
an address error will occur if an attempt is made to access word data starting from an add
than 2n or longword data starting from an address other than 4n. In such cases, the data ac
cannot be guaranteed. The hardware stack area, pointed by the hardware stack pointer (SI
uses only longword data starting from address 4n because this area holds the program cou
status register.

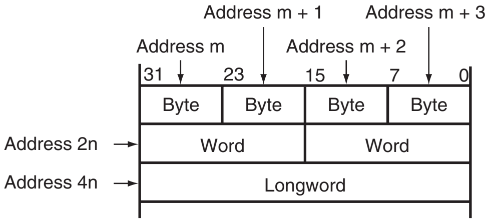


Figure 2.3 Memory Data Format

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 prog
efficiency.

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

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

**Table 2.2 Word Data Sign Extension**

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

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

| CPU in this LSI | | Description | Example of Oth | |
|---|---|---|---|---|
| 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 operati 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 branc performed according to whether the result is True or False. Processing speed has been im 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 Oth | |
|---|---|---|---|---|
| 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 | TRGET |
| BF | TRGET1 | When R0 < R1, a branch is made to TRGET1. | BLT | TRGET |
| 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 longw immediate data is not placed in the instruction code. It is placed in a table in memory. Th memory is accessed with the MOV immediate data instruction using PC relative addressi with displacement.

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

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

**Table 2.6 Access to Absolute Address**

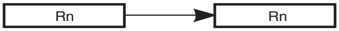
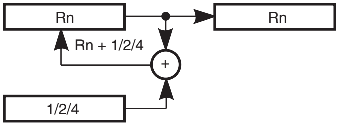
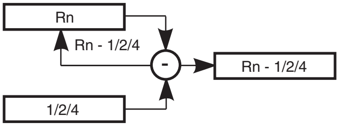
| Type | CPU in this LSI | Example of Oth |
|---|---|---|
| Absolute address | MOV.L @(disp,PC),R1<br>MOV.B @R1,R0<br>........<br>.DATA.L H'12345678 | MOV.B @H'123 |

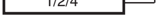
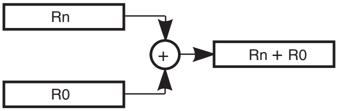
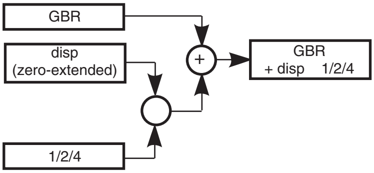
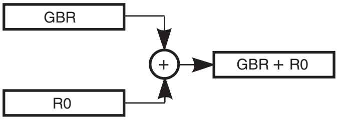
Note: Immediate data is referenced by @(disp,PC).

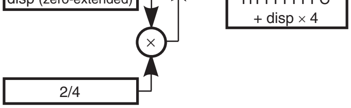
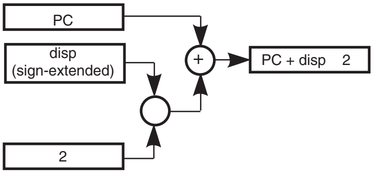
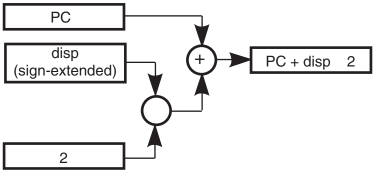
**16-Bit/32-Bit Displacement:** When data is accessed using the 16- or 32-bit displacemen addressing mode, the displacement value is placed in a table in memory beforehand. Usi method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data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

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 | Calculati 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 inst<br>executio<br>Byte: Rn<br>Word: Rr<br>Longwor<br>→ 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<br>Word: Rr<br>Longwor<br>→ Rn<br>(Instructic<br>executed<br>after calc |

| | | | |
|---|---|---|---|
| | |  | |
| Index register indirect | @(R0, Rn) | Effective address is sum of register Rn and R0 contents.<br> | Rn + R |
| 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.<br> | Byte: G<br>Word: G × 2<br>Longwo disp × 4 |
| Index GBR indirect | @(R0, GBR) | Effective address is sum of register GBR and R0 contents.<br> | GBR + |

<table>
<tr><td></td><td></td><td> </td><td></td></tr>
<tr><td>PC relative</td><td>disp:8</td><td>Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2. </td><td>PC + dis</td></tr>
<tr><td></td><td>disp:12</td><td>Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2. </td><td>PC + dis</td></tr>
</table>

| | | |
|---|---|---|
| #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 destina operands. The meaning of the operands depends on the instruction code. The following are used in the table.

`xxxx:` Instruction code

`mmmm:` Source register

`nnnn:` Destination register

`iiii:` Immediate data

`dddd:` Displacement

| | | | |
|---|---|---|---|
| | Control register or system register | nnnn: pre-decrement register indirect | STC.L SR,@-R |
| 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+,S |
| | mmmm: register indirect | — | JMP @Rm |
| | PC relative using Rm | — | BRAF Rm |

| | | | |
|---|---|---|---|
| | nnnn: * post-increment register indirect (multiply-and-accumulate operation) | | |
| | mmmm: post-increment register indirect | nnnn: register direct | MOV.L @Rm- |
| | mmmm: register direct | nnnn: pre-decrement register indirect | MOV.L Rm,@ |
| | mmmm: register direct | nnnn: index register indirect | MOV.L Rm,@ |
| md type<br>15 0<br>xxxx xxxx mmmm dddd | mmmmdddd: register indirect with displacement | R0 (register direct) | MOV.B @(dis |
| nd4 type<br>15 0<br>xxxx xxxx nnnn dddd | R0 (register direct) | nnnndddd: register indirect with displacement | MOV.B R0,@ |
| nmd type<br>15 0<br>xxxx nnnn mmmm dddd | mmmm: register direct | nnnndddd: register indirect with displacement | MOV.L Rm,@ |
| | mmmmdddd: register indirect with displacement | nnnn: register direct | MOV.L @(dis |

| | | | |
|---|---|---|---|
| | — | dddddddd:<br>PC relative | BF label |
| d12 type<br>15 0<br>xxxx dddd dddd dddd | — | dddddddddddd:<br>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, |
| i type<br>15 0<br>xxxx xxxx iiii iiii | iiiiiiii:<br>immediate | Index GBR indirect | AND.B #imm,@ |
| | iiiiiiii:<br>immediate | R0 (register direct) | AND #imm,R0 |
| | iiiiiiii:<br>immediate | — | TRAPA #imm |
| ni type<br>15 0<br>xxxx nnnn iiii iiii | iiiiiiii:<br>immediate | nnnn: register direct | ADD #imm,Rn |

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

| | | | | |
|---|---|---|---|---|
| instructions | | | Immediate data transfer<br>Peripheral module data transfer<br>Structure data transfer | |
| | | 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 | |

| | | | | |
|---|---|---|---|---|
| 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 | |

| | | | | |
|---|---|---|---|---|
| | | 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 | | | 14 |

OP: Operation code
Sz: Size
SRC: Source
DEST: Destination

Rm: Source register

Rn: Destination register

imm: Immediate data

disp: Displacement*2

nnnn: Destination register
0000: R0
0001: R1
·········
1111: R15

iiii: Immediate data

dddd: Displacement

(xx): Memory operand

M/Q/T: Flag bits in SR

&: Logical AND of each bit

|: Logical OR of each bit

^: Exclusive logical OR of each bit

–: Logical NOT of each bit

<<n: n-bit left shift

>>n: n-bit right shift

Notes: 1. The table shows the minimum number of execution states. In practice, the num 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 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.

|  |  |  |  |  |
|---|---|---|---|---|
| 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 |

| | | | | | |
|---|---|---|---|---|---|
| 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 | - |

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

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 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,<br>32 × 32 + 64 → 64 bits | 0000nnnnmmmm1111 | 2 to 5* | – |
| MAC.W | @Rm+,@Rn+ | Signed operation of (Rn) × (Rm) + MAC → MAC,<br>16 × 16 + 64 → 64 bits | 0100nnnnmmmm1111 | 2 to 4* | – |
| MUL.L | Rm,Rn | Rn × Rm → MACL<br>32 × 32 → 32 bits | 0000nnnnmmmm0111 | 2 to 5* | – |
| MULS.W | Rm,Rn | Signed operation of Rn × Rm → MAC<br>16 × 16 → 32 bits | 0010nnnnmmmm1111 | 1 to 3* | – |
| MULU.W | Rm,Rn | Unsigned operation of Rn × Rm → MAC<br>16 × 16 → 32 bits | 0010nnnnmmmm1110 | 1 to 3* | – |
| NEG | Rm,Rn | 0-Rm → Rn | 0110nnnnmmmm1011 | 1 | – |
| NEGC | Rm,Rn | 0-Rm-T → Rn,<br>Borrow → T | 0110nnnnmmmm1010 | 1 |  |
| SUB | Rm,Rn | Rn-Rm → Rn | 0011nnnnmmmm1000 | 1 | – |
| SUBC | Rm,Rn | Rn-Rm–T → Rn,<br>Borrow → T | 0011nnnnmmmm1010 | 1 |  |
| SUBV | Rm,Rn | Rn-Rm → Rn,<br>Underflow → T | 0011nnnnmmmm1011 | 1 |  |

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

| | | | | |
|---|---|---|---|---|
| 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;<br>1 → MSB of (Rn) | 0100nnnn00011011 | 4 |
| TST | Rm,Rn | Rn & Rm; if the result is 0, 1 → T | 0010nnnnmmmm1000 | 1 |
| TST | #imm,R0 | R0 & imm; if the result is 0, 1 → T | 11001000iiiiiiii | 1 |
| TST.B | #imm,@(R0,GBR) | (R0 + GBR) & imm;<br>if the result is 0, 1 → T | 11001100iiiiiiii | 3 |
| 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 |

| | | | | | |
|---|---|---|---|---|---|
| SHAL | Rn | T ← Rn ← 0 | 0100nnnn00100000 | 1 | |
| SHAR | Rn | MSB → Rn → T | 0100nnnn00100001 | 1 | |
| SHLL | Rn | T ← Rn ← 0 | 0100nnnn00000000 | 1 | |
| SHLR | Rn | 0 → Rn → T | 0100nnnn00000001 | 1 | |
| 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 | - |

| | | | | |
|---|---|---|---|---|
| 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, disp × 2 + PC → PC | 1010dddddddddddd | 2 |
| BRAF | Rm | Delayed branch, Rm + PC → PC | 0000mmmm00100011 | 2 |
| BSR | label | Delayed branch, PC → PR, disp × 2 + PC → PC | 1011dddddddddddd | 2 |
| BSRF | Rm | Delayed branch, PC → PR, Rm + PC → PC | 0000mmmm00000011 | 2 |
| JMP | @Rm | Delayed branch, Rm → PC | 0100mmmm00101011 | 2 |
| JSR | @Rm | Delayed branch, PC → PR, Rm → PC | 0100mmmm00001011 | 2 |
| RTS | | Delayed branch, PR → PC | 0000000000001011 | 2 |

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

| | | | | |
|---|---|---|---|---|
| LDC Rm,VBR | Rm → VBR | 0100mmmm00101110 | 1 | |
| LDC.L @Rm+,SR | (Rm) → SR, Rm + 4 → Rm | 0100mmmm00000111 | 8 | |
| 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,<br>Stack area → PC/SR | 0000000000101011 | 5 | - |
| SETT | 1 → T | 0000000000011000 | 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 | - |

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 numb 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 used by the instruction immediately after the load instruction.

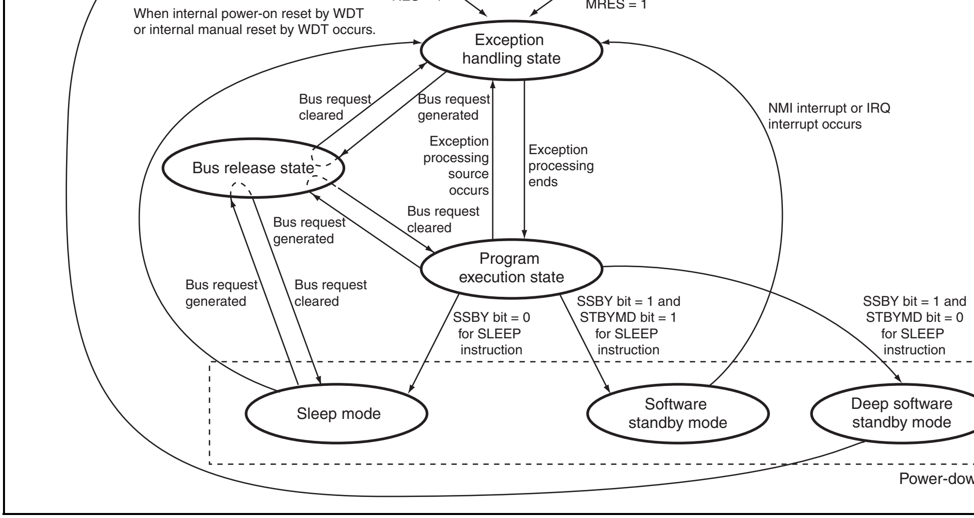


**Figure 2.4 Transitions between Processing States**

by SP. The start address of an exception handling routine is fetched from the excepti 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 CP 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 h requested them.

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

**Table 3.1 Selection of Operating Modes**

| | Pin Setting | | | | | Bus Width of CS0 S | |
|---|---|---|---|---|---|---|---|
| **Mode No.** | **FWE** | **MD1** | **MD0*1** | **Mode Name** | **On-Chip ROM** | **SH7131/SH7136** | **SH713** |
| 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 foll
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.

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

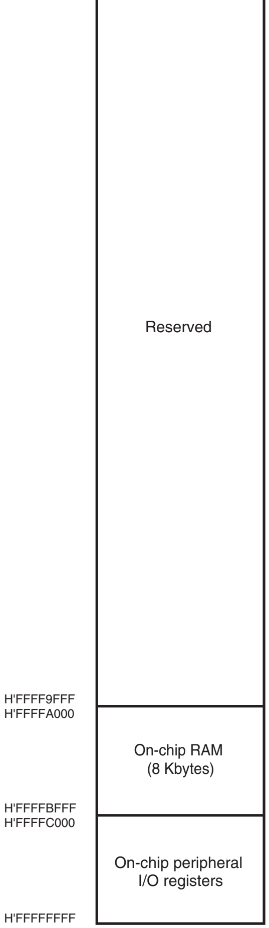


**Figure 3.1 Address Map for Each Operating Mode in SH7131 (128-Kbyte Flash Memory Version)**

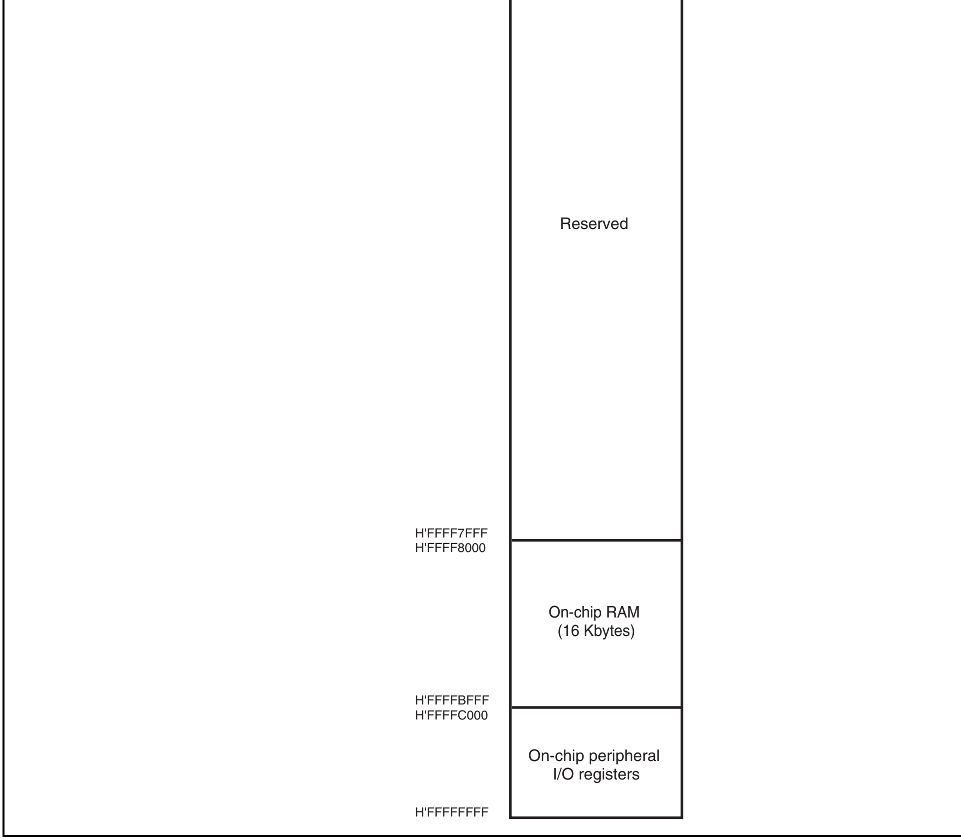


**Figure 3.2 Address Map for Each Operating Mode in SH7131 and SH713**
**(256-Kbyte Flash Memory Version)**

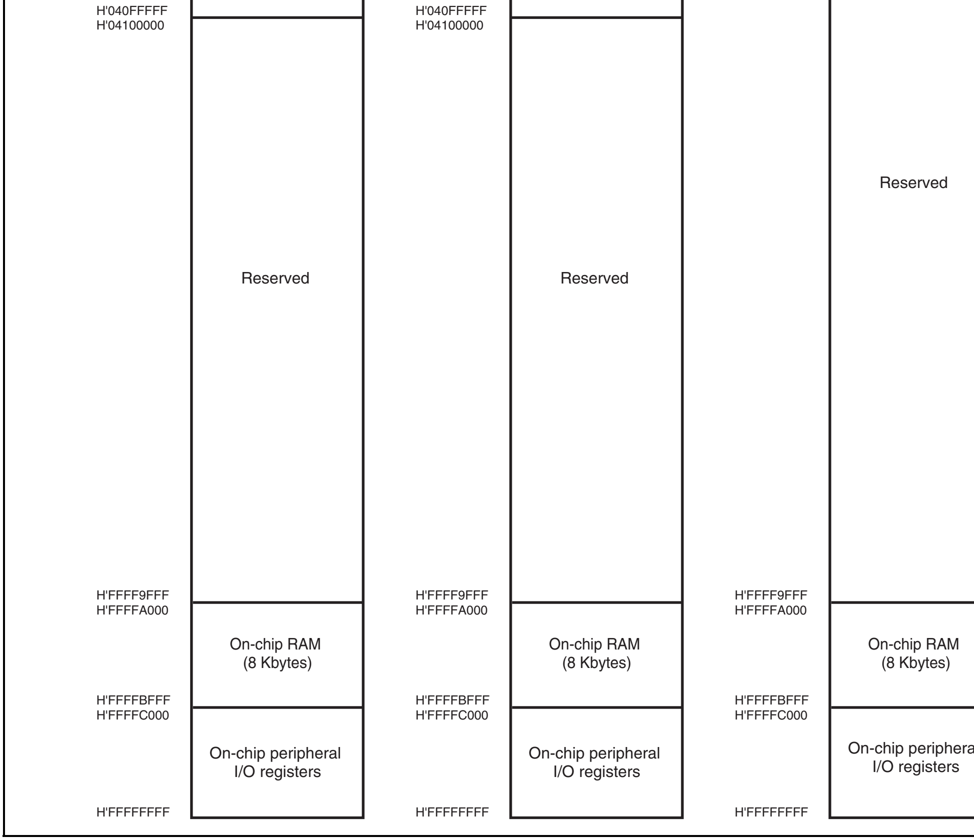


**Figure 3.3 Address Map for Each Operating Mode in SH7132 (128-Kbyte Flash Memory Version)**

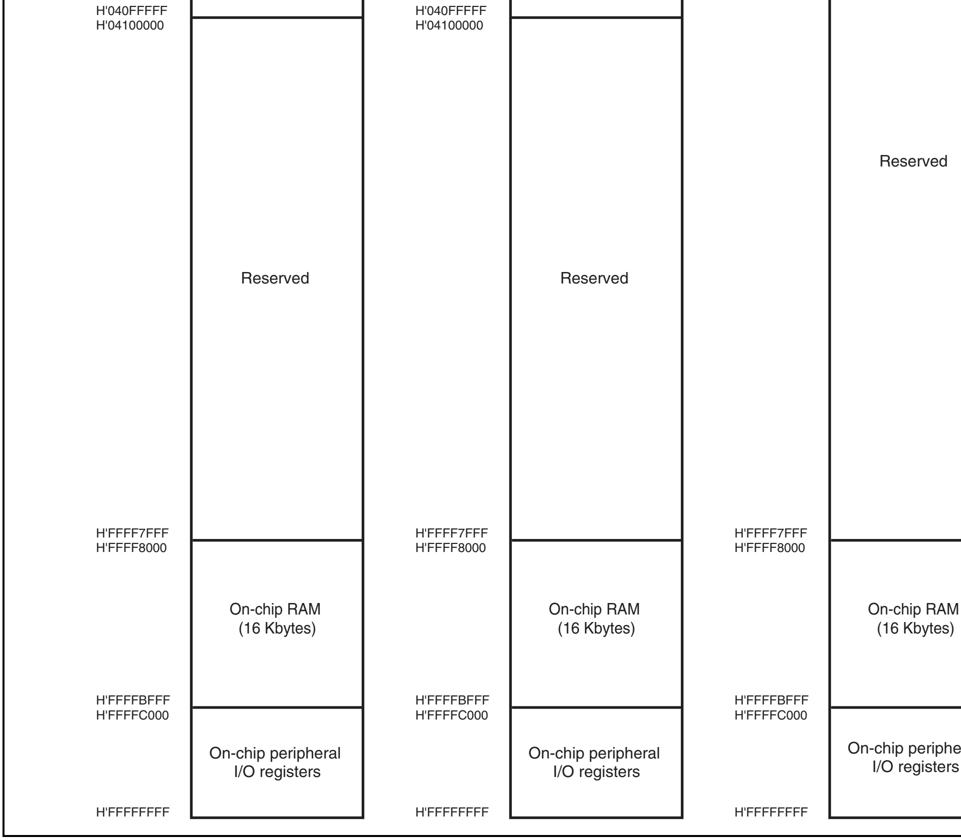
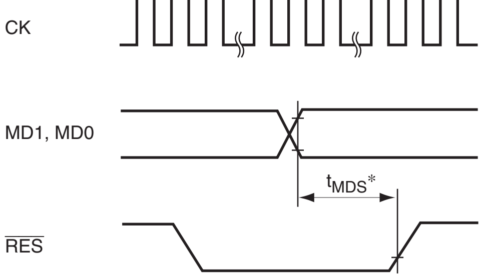


**Figure 3.4 Address Map for Each Operating Mode in SH7132 and SH713**
**(256-Kbyte Flash Memory Version)**



Note: * See section 26.3.2, Control Signal Timing.

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

a bus clock (Bϕ = CK) for the external bus interface, a MTU2S clock (MIϕ) for the o
MTU2S module; and a MTU2 clock (MPϕ) for the on-chip MTU2 module.

- Frequency change function

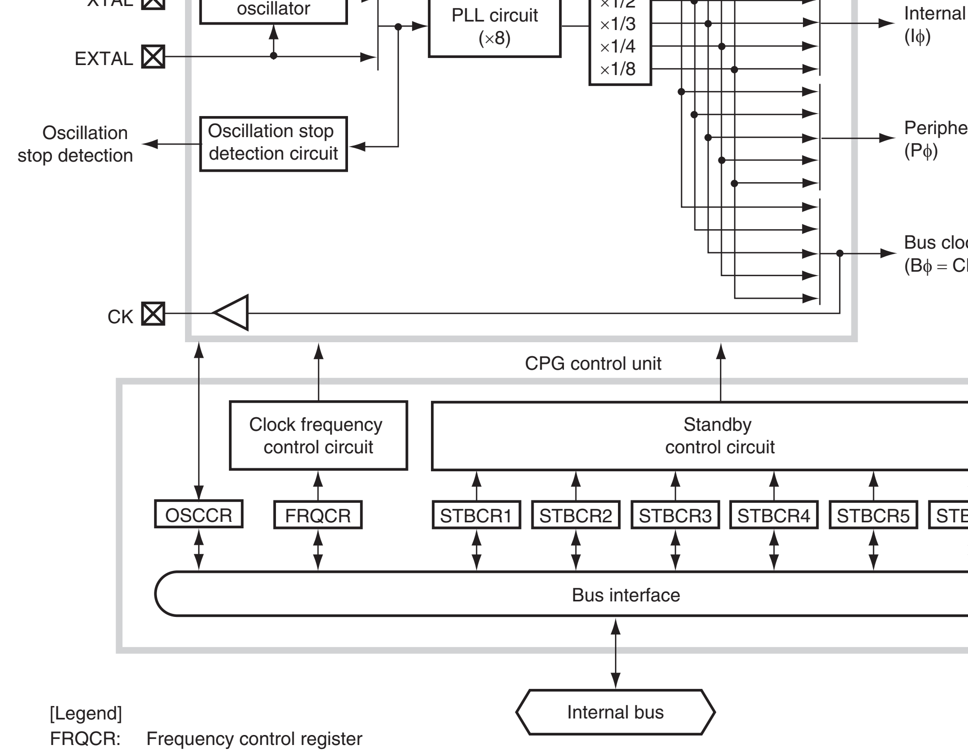
  Frequencies of the internal clock (Iϕ), bus clock (Bϕ), peripheral clock (Pϕ), MTU2S (MIϕ), and MTU2 clock (MPϕ) can be changed independently using the divider circu the CPG. Frequencies are changed by software using the frequency control register ( setting.

- Power-down mode control

  The clock can be stopped in sleep mode and standby mode and specific modules can 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 automatically placed in the high-impedance state.



[Legend]

| | |
|---|---|
| FRQCR: | Frequency control register |
| OSCCR: | Oscillation stop detection control register |
| STBCR1: | Standby control register 1 |
| STBCR2: | Standby control register 2 |
| STBCR3: | Standby control register 3 |
| STBCR4: | Standby control register 4 |
| STBCR5: | Standby control register 5 |
| STBCR6: | Standby control register 6 |

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

PLL circuit. The division ratio should be specified in the frequency control register (FR

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

**Clock Frequency Control Circuit:** The clock frequency control circuit controls the clo 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 c circuit and other modules in sleep or standby mode.

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

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

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

|  |  |  |  |
|---|---|---|---|
|  |  |  | A/D |
|  |  |  | CMT |
|  |  |  | WDT |
| Bus clock (Bϕ) | BSC | MTU2 clock (MPϕ) | MTU2 |
|  | DTC | MTU2S clock (MIϕ) | MTU2S |

Note: * SH7136 and SH7137 only.

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

made in the pin function controller (PFC). For details, refer to section 20, Pin F Controller (PFC).

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

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.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1/4 | 1/4 | 1/8 | 1/4 | 1/8 | 2 | 2 | 1 | 2 | 1 | 20 | 20 | 10 |
| 1/4 | 1/4 | 1/8 | 1/4 | 1/4 | 2 | 2 | 1 | 2 | 2 | 20 | 20 | 10 |
| 1/4 | 1/4 | 1/4 | 1/4 | 1/4 | 2 | 2 | 2 | 2 | 2 | 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 |
| 1/2 | 1/8 | 1/8 | 1/8 | 1/8 | 4 | 1 | 1 | 1 | 1 | 40 | 10 | 10 |
| 1/2 | 1/8 | 1/8 | 1/4 | 1/8 | 4 | 1 | 1 | 2 | 1 | 40 | 10 | 10 |
| 1/2 | 1/8 | 1/8 | 1/2 | 1/8 | 4 | 1 | 1 | 4 | 1 | 40 | 10 | 10 |
| 1/2 | 1/4 | 1/8 | 1/8 | 1/8 | 4 | 2 | 1 | 1 | 1 | 40 | 20 | 10 |
| 1/2 | 1/4 | 1/8 | 1/4 | 1/8 | 4 | 2 | 1 | 2 | 1 | 40 | 20 | 10 |
| 1/2 | 1/4 | 1/8 | 1/4 | 1/4 | 4 | 2 | 1 | 2 | 2 | 40 | 20 | 10 |
| 1/2 | 1/4 | 1/8 | 1/2 | 1/8 | 4 | 2 | 1 | 4 | 1 | 40 | 20 | 10 |
| 1/2 | 1/4 | 1/8 | 1/2 | 1/4 | 4 | 2 | 1 | 4 | 2 | 40 | 20 | 10 |
| 1/2 | 1/4 | 1/4 | 1/4 | 1/4 | 4 | 2 | 2 | 2 | 2 | 40 | 20 | 20 |
| 1/2 | 1/4 | 1/4 | 1/2 | 1/4 | 4 | 2 | 2 | 4 | 2 | 40 | 20 | 20 |
| 1/2 | 1/2 | 1/8 | 1/8 | 1/8 | 4 | 4 | 1 | 1 | 1 | 40 | 40 | 10 |
| 1/2 | 1/2 | 1/8 | 1/4 | 1/8 | 4 | 4 | 1 | 2 | 1 | 40 | 40 | 10 |
| 1/2 | 1/2 | 1/8 | 1/4 | 1/4 | 4 | 4 | 1 | 2 | 2 | 40 | 40 | 10 |
| 1/2 | 1/2 | 1/8 | 1/2 | 1/8 | 4 | 4 | 1 | 4 | 1 | 40 | 40 | 10 |
| 1/2 | 1/2 | 1/8 | 1/2 | 1/4 | 4 | 4 | 1 | 4 | 2 | 40 | 40 | 10 |
| 1/2 | 1/2 | 1/8 | 1/2 | 1/2 | 4 | 4 | 1 | 4 | 4 | 40 | 40 | 10 |
| 1/2 | 1/2 | 1/4 | 1/4 | 1/4 | 4 | 4 | 2 | 2 | 2 | 40 | 40 | 20 |
| 1/2 | 1/2 | 1/4 | 1/2 | 1/4 | 4 | 4 | 2 | 4 | 2 | 40 | 40 | 20 |
| 1/2 | 1/2 | 1/4 | 1/2 | 1/2 | 4 | 4 | 2 | 4 | 4 | 40 | 40 | 20 |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1/1 | 1/4 | 1/8 | 1/4 | 1/8 | 8 | 2 | 1 | 2 | 1 | 80 | 20 | 10 | 2 |
| 1/1 | 1/4 | 1/8 | 1/4 | 1/4 | 8 | 2 | 1 | 2 | 2 | 80 | 20 | 10 | 2 |
| 1/1 | 1/4 | 1/8 | 1/2 | 1/8 | 8 | 2 | 1 | 4 | 1 | 80 | 20 | 10 | 4 |
| 1/1 | 1/4 | 1/8 | 1/2 | 1/4 | 8 | 2 | 1 | 4 | 2 | 80 | 20 | 10 | 4 |
| 1/1 | 1/4 | 1/8 | 1/1 | 1/8 | 8 | 2 | 1 | 8 | 1 | 80 | 20 | 10 | 8 |
| 1/1 | 1/4 | 1/8 | 1/1 | 1/4 | 8 | 2 | 1 | 8 | 2 | 80 | 20 | 10 | 8 |
| 1/1 | 1/4 | 1/4 | 1/4 | 1/4 | 8 | 2 | 2 | 2 | 2 | 80 | 20 | 20 | 2 |
| 1/1 | 1/4 | 1/4 | 1/2 | 1/4 | 8 | 2 | 2 | 4 | 2 | 80 | 20 | 20 | 4 |
| 1/1 | 1/4 | 1/4 | 1/1 | 1/4 | 8 | 2 | 2 | 8 | 2 | 80 | 20 | 20 | 8 |
| 1/1 | 1/3 | 1/3 | 1/3 | 1/3 | 8 | 8/3 | 8/3 | 8/3 | 8/3 | 80 | 26 | 26 | 2 |
| 1/1 | 1/3 | 1/3 | 1/1 | 1/3 | 8 | 8/3 | 8/3 | 8 | 8/3 | 80 | 26 | 26 | 8 |
| 1/1 | 1/2 | 1/8 | 1/8 | 1/8 | 8 | 4 | 1 | 1 | 1 | 80 | 40 | 10 | 1 |
| 1/1 | 1/2 | 1/8 | 1/4 | 1/8 | 8 | 4 | 1 | 2 | 1 | 80 | 40 | 10 | 2 |
| 1/1 | 1/2 | 1/8 | 1/4 | 1/4 | 8 | 4 | 1 | 2 | 2 | 80 | 40 | 10 | 2 |
| 1/1 | 1/2 | 1/8 | 1/2 | 1/8 | 8 | 4 | 1 | 4 | 1 | 80 | 40 | 10 | 4 |
| 1/1 | 1/2 | 1/8 | 1/2 | 1/4 | 8 | 4 | 1 | 4 | 2 | 80 | 40 | 10 | 4 |
| 1/1 | 1/2 | 1/8 | 1/2 | 1/2 | 8 | 4 | 1 | 4 | 4 | 80 | 40 | 10 | 4 |
| 1/1 | 1/2 | 1/8 | 1/1 | 1/8 | 8 | 4 | 1 | 8 | 1 | 80 | 40 | 10 | 8 |
| 1/1 | 1/2 | 1/8 | 1/1 | 1/4 | 8 | 4 | 1 | 8 | 2 | 80 | 40 | 10 | 8 |
| 1/1 | 1/2 | 1/8 | 1/1 | 1/2 | 8 | 4 | 1 | 8 | 4 | 80 | 40 | 10 | 8 |
| 1/1 | 1/2 | 1/4 | 1/4 | 1/4 | 8 | 4 | 2 | 2 | 2 | 80 | 40 | 20 | 2 |
| 1/1 | 1/2 | 1/4 | 1/2 | 1/4 | 8 | 4 | 2 | 4 | 2 | 80 | 40 | 20 | 4 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1/1 | 1/1 | 1/4 | 1/2 | 1/4 | 8 | 8 | 2 | 4 | 2 | 40 | 40 | 10 |
| 1/1 | 1/1 | 1/4 | 1/2 | 1/2 | 8 | 8 | 2 | 4 | 4 | 40 | 40 | 10 |
| 1/1 | 1/1 | 1/4 | 1/1 | 1/4 | 8 | 8 | 2 | 8 | 2 | 40 | 40 | 10 |
| 1/1 | 1/1 | 1/4 | 1/1 | 1/2 | 8 | 8 | 2 | 8 | 4 | 40 | 40 | 10 |
| 1/1 | 1/1 | 1/4 | 1/1 | 1/1 | 8 | 8 | 2 | 8 | 8 | 40 | 40 | 10 |
| 1/1 | 1/1 | 1/3 | 1/3 | 1/3 | 8 | 8 | 8/3 | 8/3 | 8/3 | 40 | 40 | 13 |
| 1/1 | 1/1 | 1/3 | 1/1 | 1/3 | 8 | 8 | 8/3 | 8 | 8/3 | 40 | 40 | 13 |
| 1/1 | 1/1 | 1/3 | 1/1 | 1/1 | 8 | 8 | 8/3 | 8 | 8 | 40 | 40 | 13 |
| 1/1 | 1/1 | 1/2 | 1/2 | 1/2 | 8 | 8 | 4 | 4 | 4 | 40 | 40 | 20 |
| 1/1 | 1/1 | 1/2 | 1/1 | 1/2 | 8 | 8 | 4 | 8 | 4 | 40 | 40 | 20 |
| 1/1 | 1/1 | 1/2 | 1/1 | 1/1 | 8 | 8 | 4 | 8 | 8 | 40 | 40 | 20 |
| 1/1 | 1/1 | 1/1 | 1/1 | 1/1 | 8 | 8 | 8 | 8 | 8 | 40 | 40 | 40 |

division ratio of the divider. The resultant frequency must be a maximum of 40
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 crystal resonator or EXTAL pin, the multiplication ratio (×8) of the PLL circuit, a division ratio of the divider. The resultant frequency must be a maximum of 40 equal to or lower than the bus clock (Bϕ) frequency.
7. When using the MTU2S and MTU2, the MTU2S clock (MIϕ) frequency must be or lower than the internal clock (Iϕ) frequency and equal to or higher than the M clock (MPϕ) frequency. The MTU2 clock (MPϕ) frequency must be equal to or than the MTU2S clock (MIϕ) frequency and the bus clock (Bϕ) frequency, and or higher than the peripheral clock frequency (Pϕ). The MTU2S clock (MIϕ) fre and MTU2 clock (MPϕ) frequency are the product of the frequency of the input crystal resonator or EXTAL pin, the multiplication ratio (×8) of the PLL circuit, a division ratio of the divider.
8. The frequency of the CK pin is always be equal to the bus clock (Bϕ) frequency

| | | | | | |
|---|---|---|---|---|---|
| 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 internal clock (Iϕ), bus clock (Bϕ), peripheral clock (Pϕ), MTU2S clock (MIϕ), and MT (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 overflow).

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

| | | | | |
|---|---|---|---|---|
| | | | | 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 circuit. If a prohibited value is specified, subsec 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 |

| | | | | |
|---|---|---|---|---|
| | | | | 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 frequency with respect to the output frequenc circuit. If a prohibited value is specified, subse 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 (M frequency with respect to the output frequenc circuit. If a prohibited value is specified, subse 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 |

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

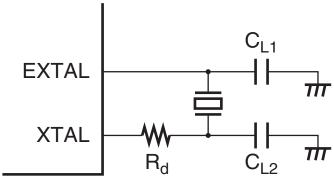
3. Set the desired values in bits IFC2 to IFC0, BFC2 to BFC0, PFC2 to PFC0, MIFC2 t and MPFC2 to MPFC0 bits. Since the frequency multiplication ratio in the PLL circu at ×8, the frequencies are determined only be selecting division ratios. When specify frequencies, satisfy the following condition: internal clock (Iϕ) ≥ bus clock (Bϕ) ≥ pe clock (Pϕ). When using the MTU2S clock and MTU2 clock, specify the frequencies the following condition: internal clock (Iϕ) ≥ MTU2S clock (MIϕ) ≥ MTU2 clock (M peripheral clock (Pϕ) and bus clock (Bϕ) ≥ MTU2 clock (MPϕ). Code to rewrite valu 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 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.



$C_{L1} = C_{L2}$ = 18 to 22 pF (Reference values)

**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 wit characteristics listed in table 4.7.

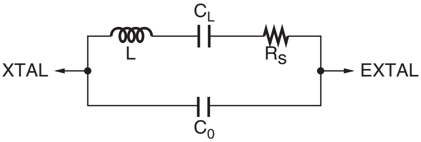


**Figure 4.3 Crystal Resonator Equivalent Circuit**

**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 |

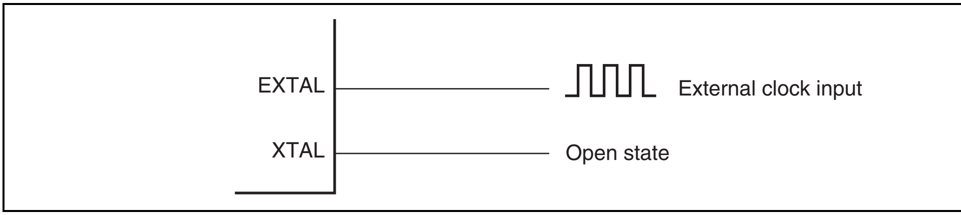


**Figure 4.4 Example of External Clock Connection**

Even in software standby mode, these pins can be placed in high-impedance state. For de
refer to appendix A, Pin States. These pins enter the normal state after software standby n
canceled. Under an abnormal condition where oscillation stops while the LSI is not in sof
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
high-current pins become unpredictable.

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

### 4.8.2 Notes on Board Design

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

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

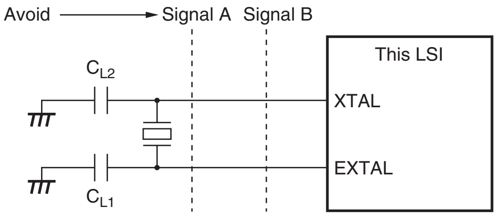
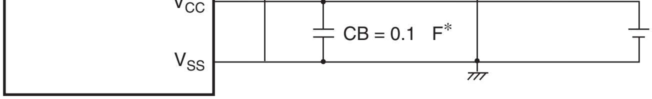


**Figure 4.5 Cautions for Oscillator Circuit Board Design**



(Recommended values are shown.)

Note: * CB and CPB are laminated ceramic type.

**Figure 4.6   Recommended External Circuitry around PLL**

### Table 6.1 Types of Exceptions and Priority

| Exception | Exception Source |
|---|---|
| Reset | Power-on reset |
| | 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 |

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

2. Instructions that change the PC value: JMP, JSR, BRA, BSR, RTS, RTE, BT, TRAPA, BF/S, BT/S, BSRF, BRAF, LDC Rm,SR, LDC.L @Rm+,SR.

3. SH7136 and SH7137 only.

| | | |
|---|---|---|
| | | WDT overflows. |
| Address error | | Detected during the instruction decode stage and started<br>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 d<br>(immediately after a delayed branch instruction) is decode |
| | Illegal slot instructions | Started when an undefined code placed at a delay slot<br>(immediately after a delayed branch instruction) or an ins<br>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
stack pointer (SP) are fetched from the exception handling vector table (PC from the addr
H'00000000 and SP from the address H'00000004 when a power-on reset. PC from the ad
H'00000008 and SP from the address H'0000000C when a manual reset.). For details, see
5.1.3, Exception Handling Vector Table. H'00000000 is then written to the vector base re
(VBR), and H'F (B'1111) is written to the interrupt mask bits (I3 to I0) in the status regist
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
saved to the stack indicated by R15. For interrupt exception handling, the interrupt priorit
written to the interrupt mask bits (I3 to I0) in SR. For address error and instruction excep
handling, bits I3 to I0 are not affected. The start address is then fetched from the exceptio
handling vector table and the program starts from that address.

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

**Table 5.3 Vector Numbers and Vector Table Address Offsets**

| Exception Handling Source | | Vector Number | Vector Table Address Of |
|---|---|---|---|
| Power-on reset | PC | 0 | H'00000000 to H'0000000 |
| | SP | 1 | H'00000004 to H'0000000 |
| Manual reset | PC | 2 | H'00000008 to H'0000000 |
| | SP | 3 | H'0000000C to H'0000000 |
| General illegal instruction | | 4 | H'00000010 to H'0000001 |
| (Reserved for system use) | | 5 | H'00000014 to H'0000001 |
| Illegal slot instruction | | 6 | H'00000018 to H'0000001 |
| (Reserved for system use) | | 7 | H'0000001C to H'0000001 |
| | | 8 | H'00000020 to H'0000002 |
| CPU address error | | 9 | H'00000024 to H'0000002 |
| DTC address error | | 10 | H'00000028 to H'0000002 |
| Interrupt | NMI | 11 | H'0000002C to H'0000002 |
| | User break*1 | 12 | H'00000030 to H'0000003 |
| (Reserved for system use) | | 13 | H'00000034 to H'0000003 |
| | | : | : |
| | | 31 | H'0000007C to H'0000007 |
| Trap instruction (user vector) | | 32 | H'00000080 to H'0000008 |
| | | : | : |
| | | 63 | H'000000FC to H'000000F |

| | | |
|---|---|---|
| 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 p 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 offse<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.

| | Reset State | | | Internal State | | |
|---|---|---|---|---|---|---|
| Type | $\overline{\text{RES}}$ | WDT Overflow | $\overline{\text{MRES}}$ | CPU, INTC | On-Chip Peripheral Module | POI I/O |
| Power-on reset | Low | — | — | Initialized | Initialized | Initi |
| | High | Overflow | High | Initialized | Initialized | Initi |
| Manual reset | High | Not overflowed | Low | Initialized | Not initialized | Not |

### 5.2.2 Power-On Reset

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

In the power-on reset state, power-on reset exception handling starts when driving the R 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 fro exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vec
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits 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, th program starts.

If a reset caused by the signal input on the RES pin and a reset caused by a WDT overflo
simultaneously, the $\overline{\text{RES}}$ pin reset has priority, and the WOVF bit in WTCSR is cleared t
When the power-on reset exception handling caused by the WDT is started, the CPU ope
follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vect
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits ( 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, 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 man
state. To reliably reset the LSI, the $\overline{\text{MRES}}$ pin should be kept at low for at least the durati
oscillation settling time that is set in WDT when in software standby mode (when the clo
halted) or at least 20 $t_{cyc}$ when the clock is operating. During manual reset, the CPU inter
is initialized. Registers of on-chip peripheral modules are not initialized. When the LSI e
manual reset status in the middle of a bus cycle, manual reset exception processing does
until the bus cycle has ended. Thus, manual resets do not abort bus cycles. However, onc
is driven low, hold the low level until the CPU becomes to be a manual reset mode after
cycle ends. (Keep at low level for at least the longest bus cycle). See appendix A, Pin Sta
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 i
kept low for a set period of time and then returned to high. The CPU will then operate in
procedures as described for power-on resets.

| | | | |
|---|---|---|---|
| Instruction fetch | CPU | Instruction fetched from even address | None (norma |
| | | Instruction fetched from odd address | Address erro |
| | | Instruction fetched from a space other than on-chip peripheral module space | None (norma |
| | | Instruction fetched from on-chip peripheral module space | Address erro |
| | | Instruction fetched from external memory space in single chip mode | Address erro |
| Data read/write | CPU or DTC | Word data accessed from even address | None (norma |
| | | Word data accessed from odd address | Address erro |
| | | Longword data accessed from a longword boundary | None (norma |
| | | Longword data accessed from other than a long-word boundary | Address erro |
| | | Byte or word data accessed in on-chip peripheral module space | None (norma |
| | | Longword data accessed in 16-bit on-chip peripheral module space | None (norma |
| | | Longword data accessed in 8-bit on-chip peripheral module space | None (norma |
| | | External memory space accessed when in single chip mode | Address erro |

3. The start address of the exception handling routine is fetched from the exception hand vector table that corresponds to the generated address error, and the program starts ex from that address. This branch is not a delayed branch.

| | | |
|---|---|---|
| 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. details on vector numbers and vector table address offsets, see table 6.3 in section 6, Inte Controller (INTC).

priority levels that can be set are 0 to 15. Level 16 cannot be set. For details on IPRA, IP IPRF, and IPRH to IPRM, see section 6.3.4, Interrupt Priority Registers A, D to F, and H (IPRA, IPRD to IPRF, and IPRH to IPRM).

**Table 5.8 Interrupt Priority**

| Type | Priority Level | Comment |
|---|---|---|
| NMI | 16 | Fixed priority level. Cannot be maske |
| User break* | 15 | Fixed priority level. Can be masked. |
| IRQ | 0 to 15 | Set with interrupt priority registers A, D and H to M (IPRA, IPRD to IPRF, and 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. NM always accepted, but other interrupts are only accepted if they have a priority level higher 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 CPU saves SR and the program counter (PC) to the stack. The priority level of the accept interrupt is written to bits I3 to I0 in SR. Although the priority level of the NMI is 16, the in bits I3 to I0 is H'F (level 15). Next, the start address of the exception handling routine from the exception handling vector table for the accepted interrupt, and program executio branches to that address and the program starts. For details on the interrupt exception han section 6.6, Interrupt Operation.

| | | |
|---|---|---|
| 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, BRA, BSR, RTS, RTE, BF/S, BT/S BRAF<br>Instructions that changes the PC v JSR, BRA, BSR, RTS, RTE, BT, B BF/S, BT/S, BSRF, BRAF, LDC Rr 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'F0 H'FFFF are decoded.

### 5.5.2 Trap Instructions

When a TRAPA instruction is executed, the trap instruction exception handling starts. T 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 addr instruction to be executed after the TRAPA instruction.
3. The CPU reads the start address of the exception handling routine from the exceptio vector table that corresponds to the vector number specified in the TRAPA instructic program execution branches to that address, and then the program starts. This branch delayed branch.

rewrites the PC.

3. The start address of the exception handling routine is fetched from the exception hand vector table that corresponds to the exception that occurred. Program execution branc 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 The CPU handles the general illegal instructions in the same procedures as in the illegal s instructions. Unlike processing of illegal slot instructions, however, the program counter is stacked is the start address of the undefined code.

| Occurrence Timing | Address<br>Error | Illegal<br>Instruction | Slot Illegal<br>Instruction | Trap<br>Instruction | I |
|---|---|---|---|---|---|
| Instruction in delay slot | ×*[2] | — | ×*[2] | — | × |
| Immediately after interrupt disabled instruction*[1] | √ | √ | √ | √ | × |

[Legend]

√: Accepted

×: Not accepted

—: Does not occur

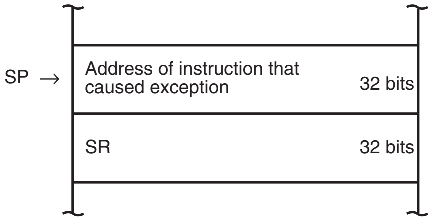
Notes: 1. Interrupt disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, a

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

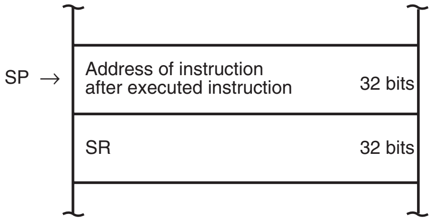
3. An exception is accepted after a delayed branch (between instructions in the and the branch destination).

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

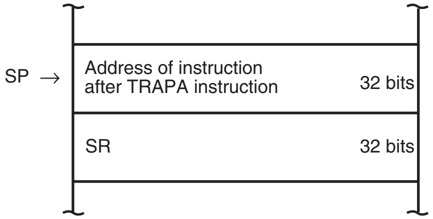
Address error (other than above)

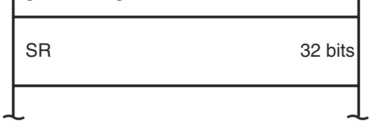


Interrupt



Trap instruction




SR
32 bits

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

When the SP value is not a multiple of 4, an address error will occur when stacking for ex handling (interrupts, etc.) and address error exception handling will start after the first exc handling is ended. Address errors will also occur in the stacking for this address error exc handling. To ensure that address error exception handling does not go into an endless loo address errors are accepted at that point. This allows program control to be passed to the routine for address error exception and enables error processing.

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

**Compiler**

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

**Real-time OS for μITRON specifications**

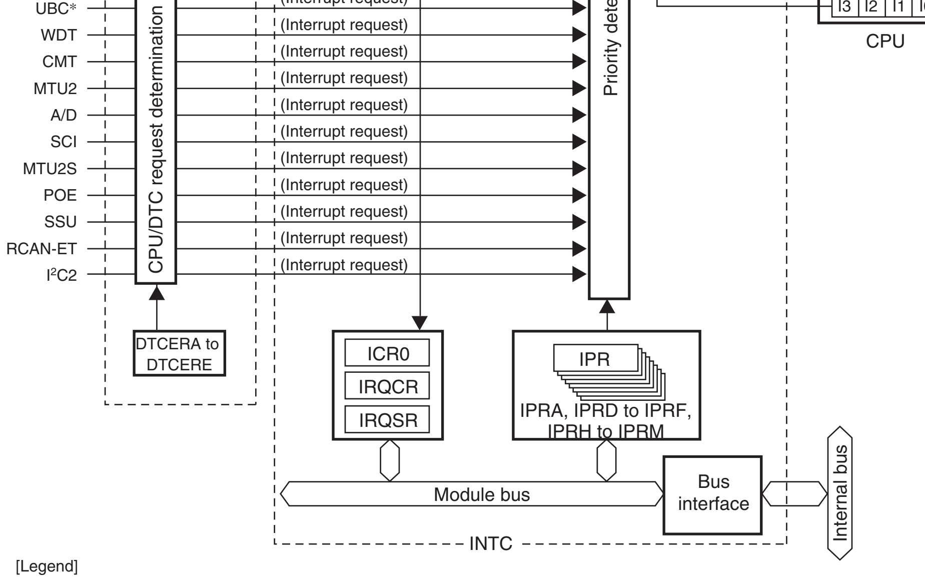
1. HI7000/4, HI-SH7

   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 illegal instruction exception handling in this LSI.
3. Others

   The slot illegal instruction exception handling may be generated in this LSI in a case instruction is described in assembler or when the middleware of the object is introdu

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



[Legend]

| | | | |
|---|---|---|---|
| UBC: | User break controller | ICR0: | Interrupt control register 0 |
| WDT: | Watchdog timer | IRQCR: | IRQ control register |
| CMT: | Compare match timer | IRQSR: | IRQ status register |
| SCI: | Serial communication interface | IPRA, IPRD to IPRF, | |
| MTU2: | Multi-function timer pulse unit 2 | IPRH to IPRM: | Interrupt priority registers A, D to F, a |
| MTU2S: | Multi-function timer pulse unit 2S | SR: | Status register |
| A/D: | A/D converter | DTCERA to DTCERE: | DTC enable registers A to E |
| POE: | Port output enable | | |
| DTC: | Data transfer controller | | |
| SSU: | Synchronous serial communication unit | | |
| RCAN-ET: | Controller area network | | |
| I$^2$C2: | I$^2$C bus interface 2 | | |

Note: * SH7136 and SH7137 only.

**Figure 6.1 Block Diagram of INTC**

| | | | |
|---|---|---|---|
| Interrupt request output pin | IRQOUT | Output | Output of notification signal whe interrupt has occurred |

| | | | | | |
|---|---|---|---|---|---|
| IRQ status register | IRQSR | R/W | H'Fx00 | H'FFFFE904 | 8, 1 |
| Interrupt priority register A | IPRA | R/W | H'0000 | H'FFFFE906 | 8, 1 |
| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | NMIL | * | R | NMI Input Level<br>Indicates the state of the signal input to the N This bit can be read to determine the NMI pin 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 va always be 0. |
| 8 | NMIE | 0 | R/W | NMI Edge Select<br>0: Interrupt request is detected on the falling e NMI input<br>1: Interrupt request is detected on the rising e NMI input |
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va always be 0. |

| | | | | |
|---|---|---|---|---|
| 15 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va<br>should always be 0. |
| 7 | IRQ31S | 0 | R/W | IRQ3 Sense Select |
| 6 | IRQ30S | 0 | R/W | Set the interrupt request detection mode for p<br>00: Interrupt request is detected at the low lev IRQ3<br>01: Interrupt request is detected at the falling pin IRQ3<br>10: Interrupt request is detected at the rising e pin IRQ3<br>11: Interrupt request is detected at both the fa 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 p<br>00: Interrupt request is detected at the low lev IRQ2<br>01: Interrupt request is detected at the falling pin IRQ2<br>10: Interrupt request is detected at the rising e pin IRQ2<br>11: Interrupt request is detected at both the fa rising edges of pin IRQ2 |

| | | | | |
|---|---|---|---|---|
| | | | | 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<br>00: Interrupt request is detected at the low le IRQ0<br>01: Interrupt request is detected at the fallin pin IRQ0<br>10: Interrupt request is detected at the rising pin IRQ0<br>11: Interrupt request is detected at both the rising edges of pin IRQ0 |

| 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 va 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 va should always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | • 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 detec<br>[Setting condition]<br>Detecting the specified edge of pin IRQ3 |
| 2 | IRQ2F | 0 | R/W | Indicates the status of an IRQ2 interrupt req<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 detec<br>[Setting condition]<br>Detecting the specified edge of pin IRQ2 |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | • 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 detecte<br>[Setting condition]<br>Detecting the specified edge of pin IRQ1 |
| 0 | IRQ0F | 0 | R/W | Indicates the status of an IRQ0 interrupt requ<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 detecte<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 the level on the pin is low.

| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R/W: | R/W | R/W | 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 inte 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) |

| | | | | |
|---|---|---|---|---|
| | | | | 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 inter 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) |

0111: Priority level 7

1000: Priority level 8

1001: Priority level 9

1010: Priority level 10

1011: Priority level 11

1100: Priority level 12

1101: Priority level 13

1110: Priority level 14

1111: Priority level 15 (highest)

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

handler, the interrupt mask level bits (I3 to I0) in the status register (SR) are set to level 1

**IRQ3 to IRQ0 Interrupts:** IRQ interrupts are requested by input from pins IRQ0 to IRQ the IRQ sense select bits (IRQ31S, IRQ30S to IRQ01S, and IRQ00S) in the IRQ control (IRQCR) to select the detection mode from low level detection, falling edge detection, ris detection, and both edge detection for each pin. The priority level can be set from 0 to 15 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 while the IRQ pin is driven low. The interrupt request signal stops to be sent to the INTC IRQ pin becomes high. It is possible to confirm that an interrupt is requested by reading t 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 IN the following change on the IRQ pin is detected: from high to low in falling edge detectio from low to high in rising edge detection mode, and from low to high or from high to low edge detection mode. The IRQ interrupt request by detecting the change on the pin is held interrupt request is accepted. It is possible to confirm that an IRQ interrupt request has be detected by reading the IRQ flags (IRQ3F to IRQ0F) in the IRQ status register (IRQSR). interrupt request by detecting the change on the pin can be withdrawn by writing 0 to an after reading 1.

In the IRQ interrupt exception handling, the interrupt mask bits (I3 to I0) in the status reg (SR) are set to the priority level value of the accepted IRQ interrupt. Figure 6.2 shows the 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 p
modules.

Since a different interrupt vector is allocated to each interrupt source, the exception hanc
routine does not have to decide which interrupt has occurred. Priority levels between 0 a
be allocated to individual on-chip peripheral modules in interrupt priority registers D to
M (IPRD to IPRF and IPRH to IPRM). On-chip peripheral module interrupt exception h
sets the interrupt mask level bits (I3 to I0) in the status register (SR) to the priority level
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 se
user break controller (UBC) is satisfied. User break interrupt requests are detected by ed
held until accepted. User break interrupt exception handling sets the interrupt mask leve
to I0) in the status register (SR) to level 15. For more details on the user break interrupt,
section 7, User Break Controller (UBC).

IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely betwee
and 15 for each pin or module by setting interrupt priority registers A, D to F and H to M
IPRD to IPRF, and IPRH to IPRM). However, when interrupt sources whose priority leve
allocated with the same IPR are requested, the interrupt of the smaller vector number has
This priority cannot be changed. Priority levels of IRQ interrupts and on-chip peripheral r
interrupts are initialized to level 0 at a power-on reset. If the same priority level is allocat
or more interrupt sources and interrupts from those sources occur simultaneously, they ar
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 |
|---|---|---|---|---|
| User break*[1] | | 12 | H'00000030 | — |
| 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 | |

|  |  |  |  |  |
|---|---|---|---|---|
|  | 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 |

| | | | | |
|---|---|---|---|---|
| | 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 | |

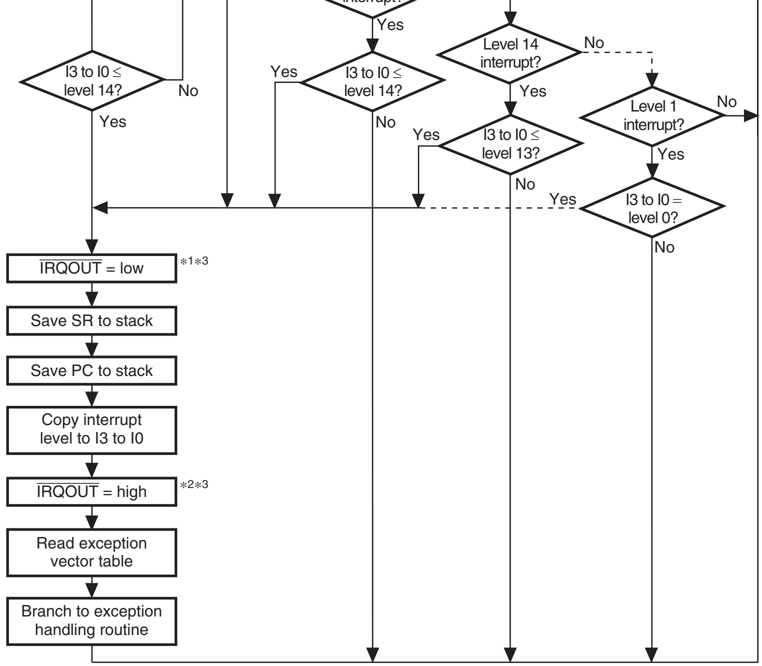
|  |  |  |
|---|---|---|
| RM1_0 |  |  |
| SLE_0 | 243 | H'000003CC |

Notes: 1. SH7136 and SH7137 only.

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

interrupt are ignored*. If interrupts that have the same priority level or interrupts with module occur simultaneously, the interrupt with the highest priority is selected accord the default priority shown in table 6.3.

3. The interrupt controller compares the priority level of the selected interrupt request w interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the priority leve selected request is equal to or less than the level set in bits I3 to I0, the request is igno the priority level of the selected request is higher than the level in bits I3 to I0, the int 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 $\overline{\text{IRQ}}$
5. The CPU detects the interrupt request sent from the interrupt controller in the decode an instruction to be executed. Instead of executing the decoded instruction, the CPU s 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 modul level is output from the $\overline{\text{IRQOUT}}$ pin. When the accepted interrupt is sensed by edge, level is output from the $\overline{\text{IRQOUT}}$ pin at the moment when the CPU starts interrupt ex processing instead of instruction execution as noted in 5. above. However, if the inter controller accepts an interrupt with a higher priority than the interrupt just to be accep $\overline{\text{IRQOUT}}$ pin holds low level.
9. The CPU reads the start address of the exception handling routine from the exception table for the accepted interrupt, branches to that address, and starts executing the prog This branch is not a delayed branch.



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 wher the CPU starts interrupt exception processing instead of instruction execution (namely, before saving SR to stac However, if the interrupt controller accepts an interrupt with a higher priority than the interrupt just to be accepte 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 sa

**Figure 6.3 Interrupt Sequence Flowchart**

Notes: 1. PC is the start address of the next instruction (instruction at the return address) after the 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 ir request until the interrupt exception handling starts and fetching of the first instruction o interrupt handling routine begins.

| | | | | | |
|---|---|---|---|---|---|
| executed by CPU | | | | | error exceptio (X = 7 × Icyc + + m3 + m4). If interrupt-mask instruction foll however, the t be even longe |
| 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 s and SR, and v 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 v are all in on-ch |
| | 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

Figures 6.5 and 6.6 show control block diagrams.

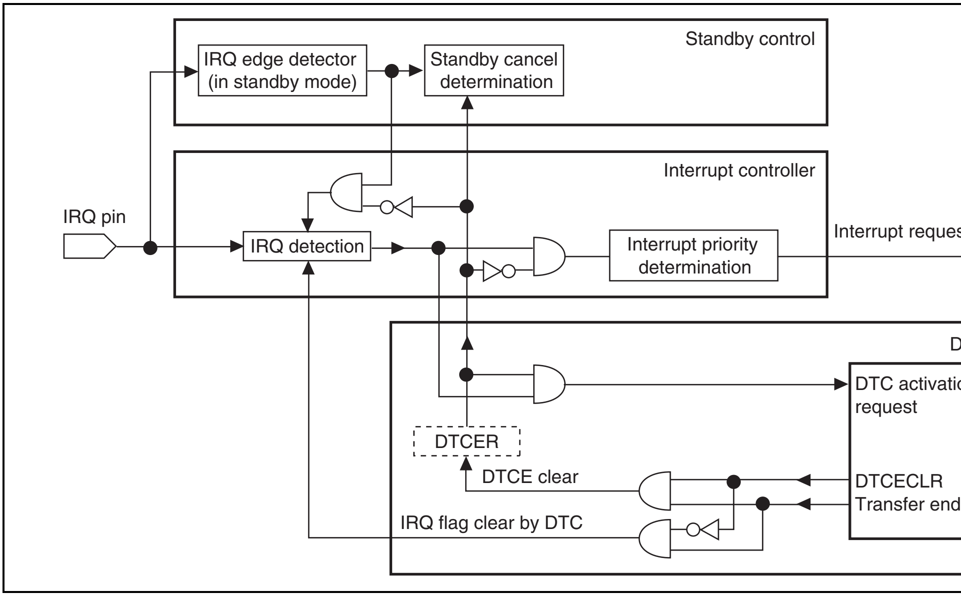


**Figure 6.5 IRQ Interrupt Control Block Diagram**

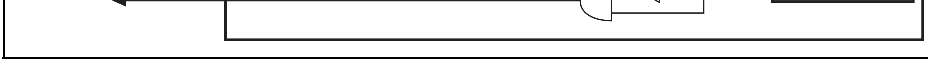


**Figure 6.6 On-Chip Module Interrupt Control Block Diagram**

### 6.8.1 Handling Interrupt Request Signals as Sources for DTC Activation and Cl 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 inter 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 t counter value. When the transfer counter value is not 0, the CPU sets the DTCE bit to allows the next data transfer. If the transfer counter value = 0, the CPU performs the n end processing in the interrupt processing routine.

### 6.8.2 Handling Interrupt Request Signals as Sources for DTC Activation, but No 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 re 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 int request is sent to the CPU.
5. The CPU performs the necessary end processing in the interrupt handling routine.

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

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 cha and B (sequential break setting: channel A and then channel B match with break con 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)) car selected.

   — Data

     32-bit maskable.

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

   — Bus cycle

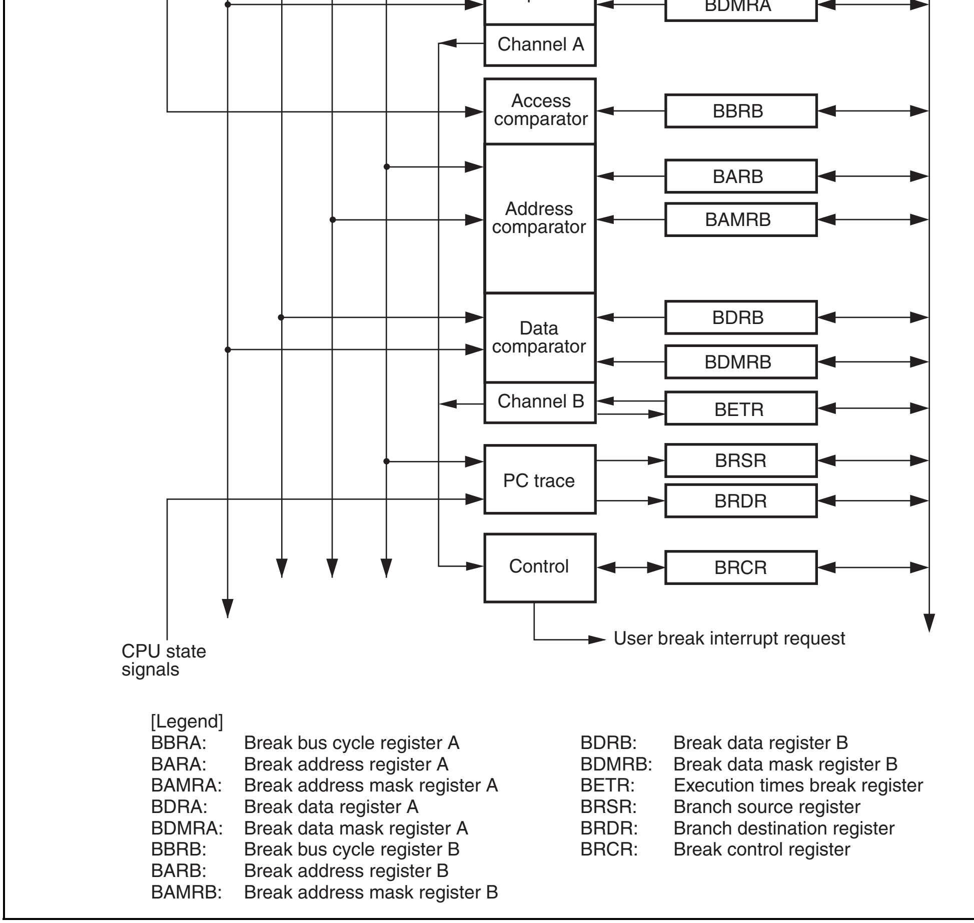
     Instruction fetch or data access

   — 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 a 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.



[Legend]

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

**Figure 7.1 Block Diagram of UBC**

| | | | | | |
|---|---|---|---|---|---|
| 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 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | 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 specifyin<br>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 bre specified by BARA.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAMA31 | BAMA30 | BAMA29 | BAMA28 | BAMA27 | BAMA26 | BAMA25 | BAMA24 | BAMA23 | BAMA22 | BAMA21 | BAMA20 | BAMA19 | BAMA18 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAMA15 | BAMA14 | BAMA13 | BAMA12 | BAMA11 | BAMA10 | BAMA9 | BAMA8 | BAMA7 | BAMA6 | BAMA5 | BAMA4 | BAMA3 | BAMA2 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

### 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 cycl bus cycle or I bus cycle, (3) instruction fetch or data access, (4) read or write, and (5) ope in the break conditions of channel A.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | CPA[2:0] | | | CDA[1:0] | | IDA[1:0] | | RWA[1:0] | |
| Initial value: | 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 |

| 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 valu 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 selecte bus cycle of the channel A break condition. Hov when the L bus is selected as the bus cycle, the 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 cor<br>x1x: Setting prohibited<br>1xx: The DTC cycle is included in the break cor |

| | | | | |
|---|---|---|---|---|
| | | | | Select the instruction fetch cycle or data acces 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<br>10: The break condition is the data access cyc<br>11: The break condition is the instruction fetch 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 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 wr |
| 1, 0 | SZA[1:0] | 00 | R/W | Operand Size Select A<br>Select the operand size of the bus cycle for th A break condition.<br>00: The break condition does not include oper<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, sp size which matches the address bour |

[Legend]

x: Don't care.

R/W: R/W R/W 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 A.<br>If the I bus is selected in BBRA, the break data set in BDA31 to BDA0.<br>If the L bus is selected in BBRA, the break data 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 bits 15 to 8 and 7 to 0 in BDRA as the break data.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | 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 cha specified by BDRA (BDA31 to BDA0).<br>0: Break data BDAn of channel A is included i break condition<br>1: Break data BDAn of channel A is masked a 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 mus bits 15 to 8 and 7 to 0 in BDMRA as the break mask data in BDRA.

| | BAB15 | BAB14 | BAB13 | BAB12 | BAB11 | BAB10 | BAB9 | BAB8 | BAB7 | BAB6 | BAB5 | BAB4 | BAB3 | BAB2 | B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | F |

| 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 conc channel B.<br>If the I bus or L bus is selected in BBRB, an IAB address is set in BAB31 to BAB0. |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | 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<br>specified by BARB (BAB31 to BAB0).<br>0: Break address BABn of channel B is includ<br>break condition<br>1: Break address BABn of channel B is maske<br>not included in the break condition<br>Note: n = 31 to 0 |

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | BDB31 to BDB0 | All 0 | R/W | Break Data Bit B<br>Stores data which specifies a break condition ir B.<br>If the I bus is selected in BBRB, the break data set in BDB31 to BDB0.<br>If the L bus is selected in BBRB, the break data 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 bits 15 to 8 and 7 to 0 in BDRB as the break data.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | 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 cha specified by BDRB (BDB31 to BDB0).<br>0: Break data BDBn of channel B is included i break condition<br>1: Break data BDBn of channel B is masked a 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 mus bits 15 to 8 and 7 to 0 in BDMRB as the break mask data in BDRB.

| Bit | Bit Name | Value | R/W | Description |
|---|---|---|---|---|
| 15 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va 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 selec the bus cycle of the channel B break conditio However, when the L bus is selected as the b the setting of the CPB2 to CPB0 bits is disabl<br>000: Condition comparison is not performed<br>xx1: The CPU cycle is included in the break c<br>x1x: Setting prohibited<br>1xx: The DTC cycle is included in the break c |
| 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 bu 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 |

|  |  |  |  |  |
|---|---|---|---|---|
| 3, 2 | RWB[1:0] | 00 | R/W | Read/Write Select B<br>Select the read cycle or write cycle as the bu<br>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 w |
| 1, 0 | SZB[1:0] | 00 | R/W | Operand Size Select B<br>Select the operand size of the bus cycle for t<br>channel B break condition.<br>00: The break condition does not include ope<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, s size which matches the address bou |

[Legend]

x: Don't care.

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 co
conditions.

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

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

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

| | | | | |
|---|---|---|---|---|
| | | | | 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<br>Note: $t_{Bcyc}$ indicates the period of one cycle<br>external bus clock (B$\phi$ = CK). |
| 19 | UBIDB | 0 | R/W | User Break Disable B<br>Enables or disables the user break interrupt<br>when the channel B break conditions are sat<br>0: User break interrupt request is enabled wh<br>conditions are satisfied<br>1: User break interrupt request is disabled wh<br>conditions are satisfied |
| 18 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value<br>always be 0. |
| 17 | UBIDA | 0 | R/W | User Break Disable A<br>Enables or disables the user break interrupt<br>when the channel A break conditions are sat<br>0: User break interrupt request is enabled wh<br>conditions are satisfied<br>1: User break interrupt request is disabled wh<br>conditions are satisfied |
| 16 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value<br>always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | set for channel B is satisfied, this flag is set to<br>order to clear this flag, write 0 into this bit.<br>0: The L bus cycle condition for channel B doe<br>match<br>1: The L bus cycle condition for channel B ma |
| 13 | SCMFDA | 0 | R/W | I Bus Cycle Condition Match Flag A<br>When the I bus cycle condition in the break co<br>set for channel A is satisfied, this flag is set to<br>order to clear this flag, write 0 into this bit.<br>0: The I bus cycle condition for channel A doe<br>match<br>1: The I bus cycle condition for channel A mat |
| 12 | SCMFDB | 0 | R/W | I Bus Cycle Condition Match Flag B<br>When the I bus cycle condition in the break co<br>set for channel B is satisfied, this flag is set to<br>order to clear this flag, write 0 into this bit.<br>0: The I bus cycle condition for channel B doe<br>match<br>1: The I bus cycle condition for channel B mat |
| 11 | PCTE | 0 | R/W | PC Trace Enable<br>0: Disables PC trace<br>1: Enables PC trace |

| | | | | |
|---|---|---|---|---|
| | | | | should always be 0. |
| 7 | DBEA | 0 | R/W | Data Break Enable A<br>Selects whether or not the data bus condition included in the break condition of channel A.<br>0: No data bus condition is included in the co channel A<br>1: The data bus condition is included in the c channel A |
| 6 | PCBB | 0 | R/W | PC Break Select B<br>Selects the break timing of the instruction fet for channel B as before or after instruction ex<br>0: PC break of channel B is set before instruc execution<br>1: PC break of channel B is set after instructi execution |
| 5 | DBEB | 0 | R/W | Data Break Enable B<br>Selects whether or not the data bus conditior included in the break condition of channel B.<br>0: No data bus condition is included in the co channel B<br>1: The data bus condition is included in the c channel B |
| 4 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | should always be 0. |
| 0 | ETBE | 0 | R/W | Number of Execution Times Break Enable<br>Enables the execution-times break condition o channel B. If this bit is 1 (break enable), a use issued when the number of break conditions n with the number of execution times that is spe BETR.<br>0: The execution-times break condition is disa channel B<br>1: The execution-times break condition is enal channel B |

| 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 va should always be 0. |
| 11 to 0 | BET[11:0] | All 0 | R/W | Number of Execution Times |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | BSA15 | BSA14 | BSA13 | BSA12 | BSA11 | BSA10 | BSA9 | BSA8 | BSA7 | BSA6 | BSA5 | BSA4 | BSA3 | BSA2 | B |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - |  |
| R/W: | 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<br>This flag bit is set to 1 when a branch occurs.<br>is cleared to 0 when BRSR is read, the setting<br>enable PC trace is made, or BRSR is initialize<br>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 va<br>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 addres |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | BDA15 | BDA14 | BDA13 | BDA12 | BDA11 | BDA10 | BDA9 | BDA8 | BDA7 | BDA6 | BDA5 | BDA4 | BDA3 | BDA2 |
| Initial value: | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| R/W: | 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 addre stored. This flag bit is set to 1 when a branch This flag is cleared to 0 when BRDR is read, setting to enable PC trace is made, or BRDF 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 va 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 a |

instruction fetch/data access select, and read/write select) are each set. No user break generated if even one of these groups is set with B'00. The respective conditions are s bits of the break control register (BRCR). Make sure to set all registers related to brea setting BBRA or BBRB.

2. When the break conditions are satisfied, the UBC issues a user break interrupt request CPU and sets the L bus condition match flag (SCMFCA or SCMFCB) and the I bus c match flag (SCMFDA or SCMFDB) for the appropriate channel.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMFI be used to check if the set conditions match or not. The matching of the conditions se but they are not reset. Before using them again, 0 must first be written to them and th flags.
4. There is a possibility that matches of the break conditions set in channels A and B oc almost at the same time. In this case, only one user break interrupt request may be sen 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 gene all bus masters that are selected by the CPA2 to CPA0 bits in BBRA or the CPB2 bits in BBRB, and compares for a condition match.
   - I bus cycles resulting from instruction fetches on the L bus by the CPU are define instruction fetch cycles on the I bus, while other bus cycles are defined as data ac 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 cycle resulting from an instruction executed by the CPU, at which instruction the to be accepted cannot be clearly defined.

break is generated before the execution of the instruction, the user break is generated point when it has become deterministic that the instruction will be executed after it i This means this feature cannot be used on instructions fetched by overrun (instructio at a branch or during an interrupt transition, but not executed). When this kind of bre condition is set for the delay slot of a delayed branch instruction, a user break is gen prior to execution of the delayed branch instruction.

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

3. When the break condition is specified so that a user break is generated after executio instruction, the instruction that has met the break condition is executed and then the is generated before the next instruction is executed. As with pre-execution user brea cannot be used with overrun fetch instructions. When this kind of break condition is delayed branch instruction and its delay slot, a user break is not generated until the p 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 i 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 bu monitored for instruction fetch cycles to detect condition match. For details, see 5 in 7.4.1, Flow of the User Break Operation.

operand size is listed in table 7.3.

**Table 7.3 Data Access Cycle Addresses and Operand Size Comparison Condition**

| Access Size | Address Compared |
|---|---|
| Longword | Compares break address register bits 31 to 2 to address bus bits 3 |
| Word | Compares break address register bits 31 to 1 to address bus bits 3 |
| Byte | Compares break address register bits 31 to 0 to address bus bits 3 |

This means that when address H'00001003 is set in the break address register (BARA BARB), for example, the bus cycle in which the break condition is satisfied is as follo (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 by specified as the operand size of the break bus cycle register (BBRA or BBRB). When values are included in break conditions, a user break is generated when the address co and data conditions both match. To specify byte data for this case, set the same data in bytes at bits 15 to 8 and bits 7 to 0 of the break data register (BDRA or BDRB) and br mask register (BDMRA or BDMRB). When word or byte is set, bits 31 to 16 of BDR BDRB and BDMRA or BDMRB are ignored.

4. If the L bus is selected, a user break is generated on ending execution of the instructio 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 e execution of the instruction following the instruction that matches the break condition the I bus is selected, the instruction at which the user break is generated cannot be det

channel B condition match has not yet occurred when a sequential break has been sp clear the SEQ bit in BRCR and channel A condition match flag to 0 by writing a 0 to

2. In sequential break specification, the L or I bus can be selected and the execution tim condition can be also specified. For example, when the execution times break condit specified, the break condition is satisfied when a channel B condition matches with 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 r saved in the stack, and the exception handling state is entered. If the L bus is specified a condition, the instruction at which the user break should occur can be clearly determined for when data is included in the break condition). If the I bus is specified as a break conc instruction at which the user break should occur cannot be clearly determined.

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

   The address of the instruction that matched the break condition is saved in the stack. instruction that matched the condition is not executed, and the user break occurs bef However when a delay slot instruction matches the condition, the address of the dela 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 condi saved in the stack. The instruction that matches the condition is executed, and the br before the next instruction is executed. However when a delayed branch instruction slot matches the condition, these instructions are executed, and the branch destinatio is saved in the stack.

the stack. If the instruction following the instruction that matches the break condition
branch instruction, the break may occur after the branch instruction or delay slot has f
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
   exception) is generated, the branch source address and branch destination address are
   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 instructio
     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 hand
     routine is saved in BRDR.
3. BRSR and BRDR have four pairs of queue structures. The top of queues is read first v
   address stored in the PC trace register is read. BRSR and BRDR share the read pointe
   BRSR and BRDR in order, the queue only shifts after BRDR is read. After switching
   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 aft
   the MSTP25 bit in STBCR5 to 0 and the AUDSRST bit in STBCR6 to 1. This setting
   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 re
   the status of FIFO is initialized by a manual reset or an AUD software reset, clear the
   in the BRCR register to 0 once, set the PCTE bit to 1, and then the PC trace can start.

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 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 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'00000
  BDMRA = H'00000000, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H
  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

BARA = H'00027128, BAMRA = H'00000000, BBRA = H'005A, BDRA = H'00000
BDMRA = H'00000000, BARB = H'00031415, BAMRB = H'00000000, BBRB = H'
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 i included in the condition)

On channel A, no user break occurs since instruction fetch is not a write cycle. On ch 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'00000
  BDMRA = H'00000000, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'
  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

BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BDRA = H'00000
BDMRA = H'00000000, BARB = H'00001000, BAMRB = H'00000000, BBRB = H
BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000001, BETR = H'00

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 exe times and before the fifth time.

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

(Example 1-6)

- Register specifications

  BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BDRA = H'0000(
  BDMRA = H'00000000, BARB = H'00008010, BAMRB = H'00000006, BBRB = H
  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

**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'123456
  BDMRA = H'FFFFFFFF, BARB = H'000ABCDE, BAMRB = H'000000FF, BBRB =
  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, wor
  from address H'00123456, or byte read from address H'00123456. On channel B, a us
  occurs when word H'A512 is written in addresses H'000ABC00 to H'000ABCFE.

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

Bus cycle: I bus (CPU cycle)/instruction fetch/read (operand size is not included in 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 in the external memory space.

On channel B, a user break occurs when byte data H'7x is written in address H'0005 external memory space by the CPU.

match occurs in another bus cycle in sequential break setting. Therefore, no user breal if a bus cycle in which an A-channel match and a channel B match occur simultaneou

4. When a user break and another exception occur at the same instruction, which has hig 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, execution-type exception is accepted, and the condition match flag is not set (see t exception in the following note). The user break will occur and the condition matc will be set only after the exception source of the re-execution-type exception has b cleared by the exception handling routine and re-execution of the same instruction 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 no 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 genera CPU address error by data access, the CPU address error takes priority over the user b 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 u break does not occur until the branch destination of the RTE instruction.

Chain transfer is only possible after data transfer has been done for the specified nu
times (i.e. when the transfer counter is 0)

- Three transfer modes

  Normal/repeat/block transfer modes selectable

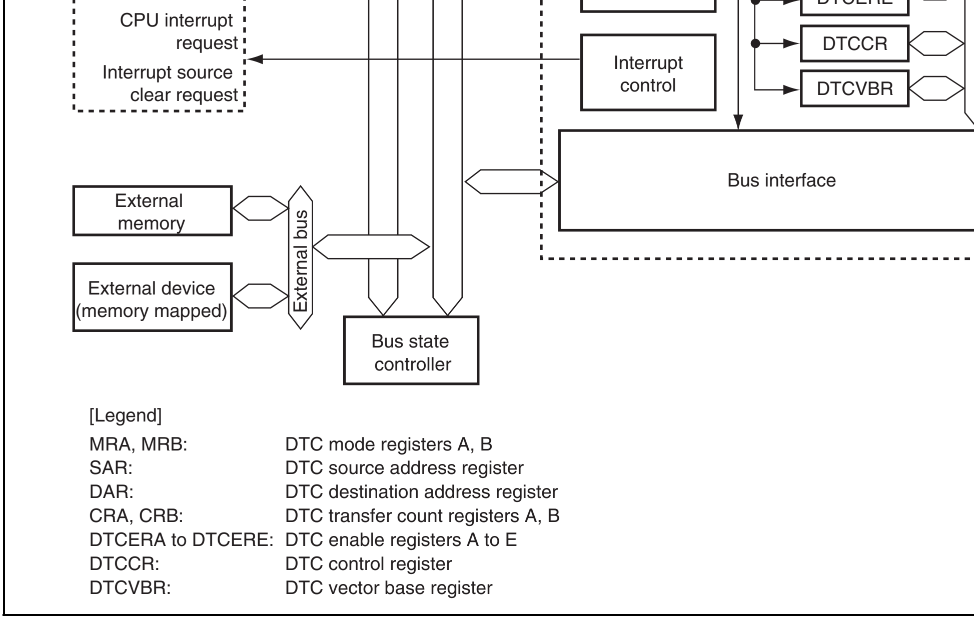
  Transfer source and destination addresses can be selected from increment/decrement
- The transfer source and destination addresses can be specified by 32 bits to select a 4
  address space directly
- Size of data for data transfer can be specified as byte, word, or longword
- A CPU interrupt can be requested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

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



[Legend]

| | |
|---|---|
| MRA, MRB: | DTC mode registers A, B |
| SAR: | DTC source address register |
| DAR: | DTC destination address register |
| CRA, CRB: | DTC transfer count registers A, B |
| DTCERA to DTCERE: | DTC enable registers A to E |
| DTCCR: | DTC control register |
| DTCVBR: | DTC vector base register |

**Figure 8.1 Block Diagram of DTC**

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

**Table 8.1 Register Configuration**

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

| | | | | |
|---|---|---|---|---|
| 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 S 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 S Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10) |

| 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 se 8.5.6, Chain Transfer. The chain transfer condit 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 foll transfer is a chain transfer, the completion chec specified transfer count is not performed and ac source flag or DTCER is not cleared.<br>0: Chain transfer every time<br>1: Chain transfer only when transfer counter = ( |

| | | | | |
|---|---|---|---|---|
| | | | | 0: Specifies the destination as repeat or block ar<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 S 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 S 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

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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
DTC.

DAR cannot be accessed directly from the CPU.

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

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

* : Undefined

eight bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-b transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and contents of CRAH are sent to CRAL when the count reaches H'00. The transfer count is 1 CRAH = CRAL = H'01, 255 when CRAH = CRAL = H'FF, and 256 when CRAH = CRA H'00.

In block transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and t eight bits (CRAL). CRAH holds the block size while CRAL functions as an 8-bit block-s counter (1 to 256 for byte, word, or longword). CRAL is decremented by 1 every time a b (word or longword) data is transferred, and the contents of CRAH are sent to CRAL whe count reaches H'00. The block size is 1 byte (word or longword) when CRAH = CRAL = 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | |
| Initial value: | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| R/W: | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

* : Undefined

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

*: Undefined

| Bit | Bit Name | 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 s 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 transfe ended<br>• When the specified number of transfers have<br>These bits are not cleared when the DISEL bit is the specified number of transfers have not ende<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 | |

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

transferring data.

0: No interrupt occurs

1: An interrupt occurs

[Clearing condition]

- When writing 0 after reading 1

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

| 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 val always be 0. |
| 11 to 0 | — | All 0 | R | |

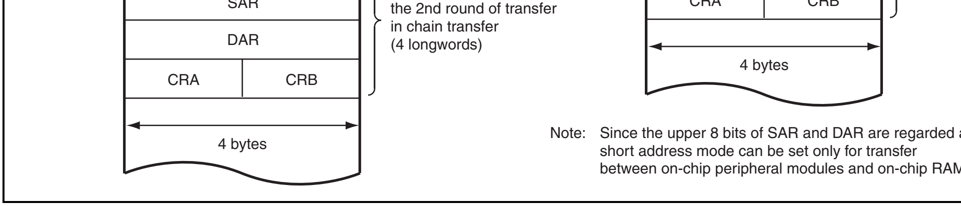
### 8.2.10 Bus Function Extending Register (BSCEHR)

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

located at the address that is a multiple of four (4n). Otherwise, the lower two bits are ign during access ([1:0] = B'00.) Transfer information located in the data area is shown in fig

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

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



Note: Since the upper 8 bits of SAR and DAR are regarded a
short address mode can be set only for transfer
between on-chip peripheral modules and on-chip RAM

**Figure 8.2 Transfer Information on Data Area**

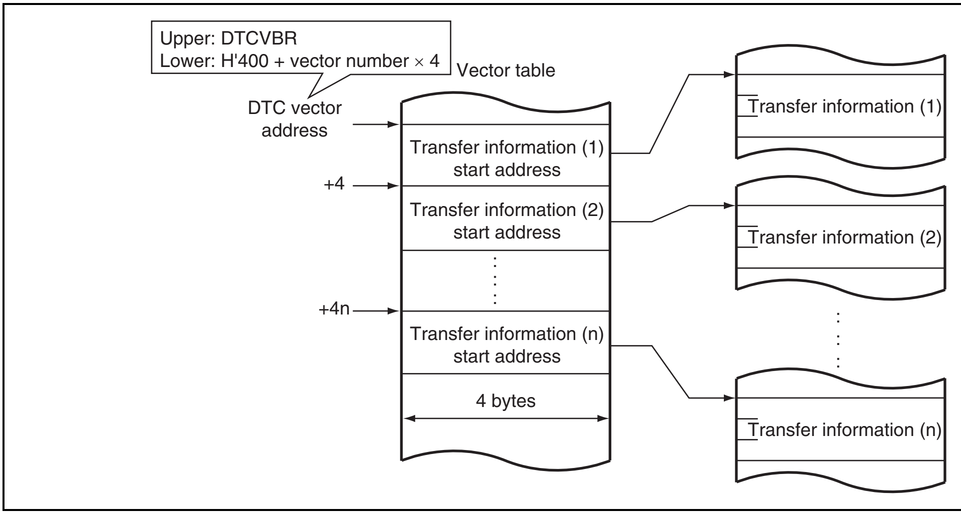


**Figure 8.3 Correspondence between DTC Vector Address and Transfer Infor**

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | TGID_4S | 171 | H'6AC | DTCERD12 | Arbitrary*2 | Arbitrary* |
| | TCIV_4S | 172 | H'6B0 | DTCERD11 | Arbitrary*2 | Arbitrary* |
| MTU2S_5 | TGIU_5S | 176 | H'6C0 | DTCERD10 | Arbitrary*2 | Arbitrary* |
| | TGIV_5S | 177 | H'6C4 | DTCERD9 | Arbitrary*2 | Arbitrary* |
| | TGIW_5S | 178 | H'6C8 | DTCERD8 | Arbitrary*2 | Arbitrary* |
| CMT_0 | CMI_0 | 184 | H'6E0 | DTCERD7 | Arbitrary*2 | Arbitrary* |
| CMT_1 | CMI_1 | 188 | H'6F0 | DTCERD6 | Arbitrary*2 | Arbitrary* |
| A/D_0 | ADI_3 | 208 | H'740 | DTCERD2 | ADDR0 to ADDR7 | Arbitrary* |
| A/D_1 | ADI_4 | 212 | H'750 | DTCERD1 | ADDR8 to ADDR15 | Arbitrary* |
| SCI_0 | RXI_0 | 217 | H'764 | DTCERE15 | SCRDR_0 | Arbitrary* |
| | TXI_0 | 218 | H'768 | DTCERE14 | Arbitrary*2 | SCTDR_ |
| SCI_1 | RXI_1 | 221 | H'774 | DTCERE13 | SCRDR_1 | Arbitrary* |
| | TXI_1 | 222 | H'778 | DTCERE12 | Arbitrary*2 | SCTDR_ |
| SCI_2 | RXI_2 | 225 | H'784 | DTCERE11 | SCRDR_2 | Arbitrary* |
| | TXI_2 | 226 | H'788 | DTCERE10 | Arbitrary*2 | SCTDR_ |
| SSU | SSRXI | 233 | H'7A4 | DTCERE7 | SSRDR0 to SSRDR3 | Arbitrary* |
| | SSTXI | 234 | H'7A8 | DTCERE6 | Arbitrary*2 | SSTDR0<br>SSTDR3 |
| $I^2C2$ | IITXI | 238 | H'7B8 | DTCERE5 | Arbitrary*2 | ICDRT |
| | IIRXI | 239 | H'7BC | DTCERE4 | ICDRR | Arbitrary* |
| RCAN-ET_0 | RM0_0 | 242 | H'7C8 | DTCERE3 | CONTROL0H to CONTROL1L*3 | Arbitrary* |

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 | Tr Co |
|---|---|---|---|
| Normal | 1 byte/word/longword | Incremented/decremented by 1, 2, or 4, or fixed | 1 |
| Repeat*[1] | 1 byte/word/longword | Incremented/decremented by 1, 2, or 4, or fixed | 1 |
| Block*[2] | Block size specified by CRAH (1 to 256 bytes/words/longwords) | Incremented/decremented by 1, 2, or 4, or fixed | 1 |

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 contir 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 w single activation (chain transfer). Setting the CHNS bit in MRB to 1 can also be made to 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 condition transfers including chain transfer (combinations for performing the second and third tra omitted).

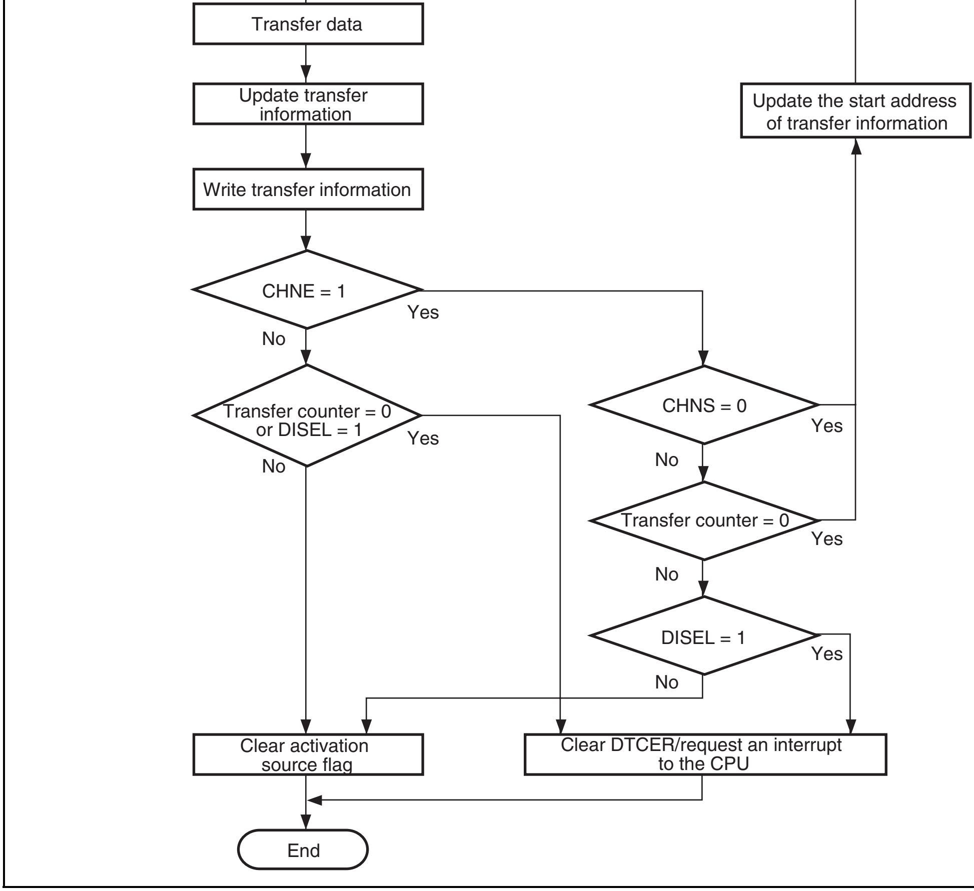

Transfer data
Update transfer information
Write transfer information
CHNE = 1
Yes
No
Transfer counter = 0 or DISEL = 1
Yes
No
Update the start address of transfer information
CHNS = 0
Yes
No
Transfer counter = 0
Yes
No
DISEL = 1
Yes
No
Clear activation source flag
Clear DTCER/request an interrupt to the CPU
End

**Figure 8.4 Flowchart of DTC Operation**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | — | — | — | 0 | — | — | 0 | Not 0 | Enc trar |
| | | | | | 0 | — | — | 0 | 0 | Enc trar Inte req |
| | | | | | 0 | — | — | 1 | — | |
| 1 | 1 | — | 0 | Not 0 | — | — | — | — | — | Enc trar |
| 1 | 1 | — | 1 | Not 0 | — | — | — | — | — | Enc trar Inte req |
| 1 | 1 | — | — | 0 | 0 | — | — | 0 | Not 0 | Enc trar |
| | | | | | 0 | — | — | 0 | 0 | Enc trar Inte req |
| | | | | | 0 | — | — | 1 | — | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | — | — | 1 | — | Ends trans Inter requ |
| 1 | 1 | — | 0 | Not 0 | — | — | — | — | — | Ends trans |
| 1 | 1 | — | 1 | Not 0 | — | — | — | — | — | Ends trans Inter requ |
| 1 | 1 | 0 | 0 | 0*2 | — | — | — | — | — | Ends trans |
| 1 | 1 | 0 | 1 | 0*2 | — | — | — | — | — | Ends trans Inter requ |
| 1 | 1 | 1 | — | 0*2 | 0 | — | — | 0 | — | Ends trans |
| | | | | | 0 | — | — | 1 | — | Ends trans Inter requ |

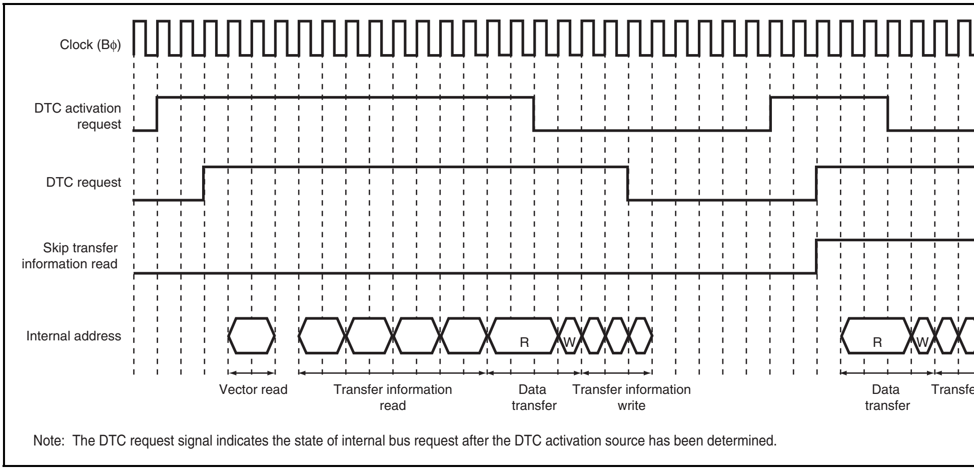
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | — | — | 0 | 0 | Enc<br>tran<br>Inte<br>req |
| | | | | | 0 | — | — | 1 | — | |
| 1 | 1 | — | 0 | — | — | — | — | — | — | Enc<br>tran |
| 1 | 1 | — | 1 | Not 0 | — | — | — | — | — | Enc<br>tran<br>Inte<br>req |
| 1 | 1 | — | 1 | 0 | 0 | — | — | 0 | Not 0 | Enc<br>tran |
| | | | | | 0 | — | — | 0 | 0 | Enc<br>tran<br>Inte<br>req |
| | | | | | 0 | — | — | 1 | — | |

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

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

cleared to 0, the stored vector number is deleted, and the updated vector table and transfe information are read at the next activation.

If the DTPR bit in the bus function extending register (BSCEHR) is set to 1, this function 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)**

| | | | |
|---|---|---|---|
| 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 un response to a single activation request. From 1 to 65,536 transfers can be specified. The source and destination addresses can be specified as incremented, decremented, or fixed 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 memo normal transfer mode.

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

| Register | Function | Written Back Value |
|---|---|---|
| SAR | Source address | Incremented/decremented/fixe |
| DAR | Destination address | Incremented/decremented/fixe |
| 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
response to a single activation request. By the DTS bit in MRB, either the source or desti
can be specified as a repeat area. From 1 to 256 transfers can be specified. When the spec
number of transfers ends, the transfer counter and address register specified as the repeat
restored to the initial state, and transfer is repeated. The other address register is then incr
decremented, or left fixed. In repeat transfer mode, the transfer counter (CRAL) is update
value specified in CRAH when CRAL becomes H'00. Thus the transfer counter value doe
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
repeat transfer mode.

| | | | |
|---|---|---|---|
| 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.

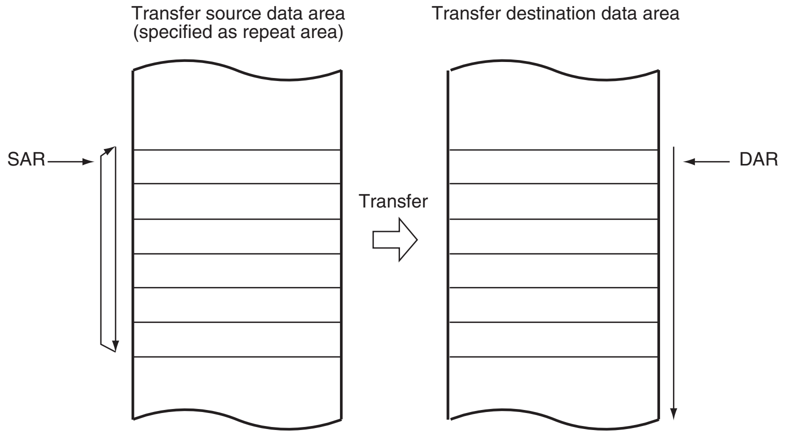

Transfer source data area
(specified as repeat area)
Transfer destination data area
SAR
Transfer
DAR

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

Table 8.8 lists the register function in block transfer mode. Figure 8.8 shows the memory 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 set to 1 enables a chain transfer only when the transfer counter reaches 0. SAR, DAR, C MRA, and MRB, which define data transfers, can be set independently. Figure 8.9 show chain transfer operation.

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not genera end of the specified number of transfers or by setting the DISEL bit to 1, and the interru 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 bi to 1 enables a chain transfer after transfer with transfer counter = 1 has been completed.

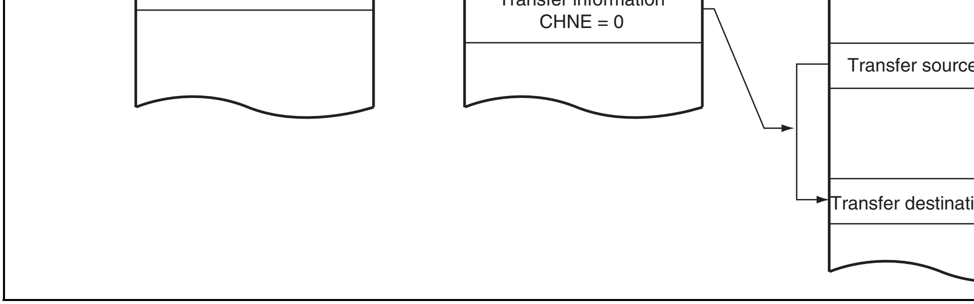
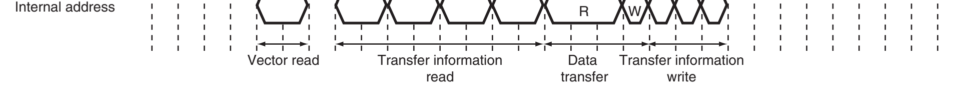
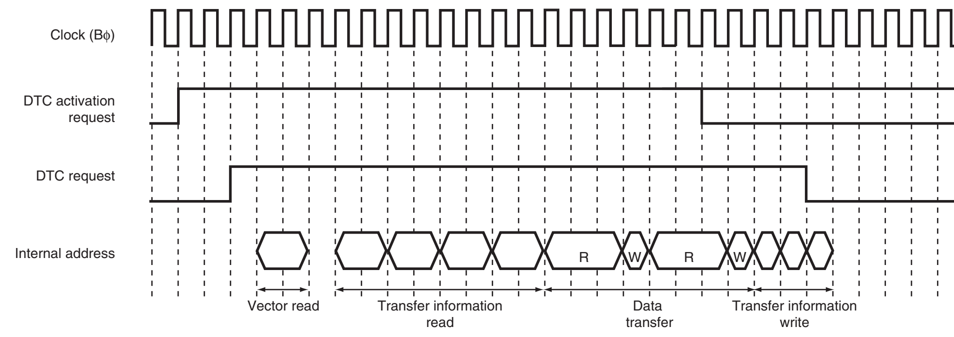


**Figure 8.9 Operation of Chain Transfer**



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

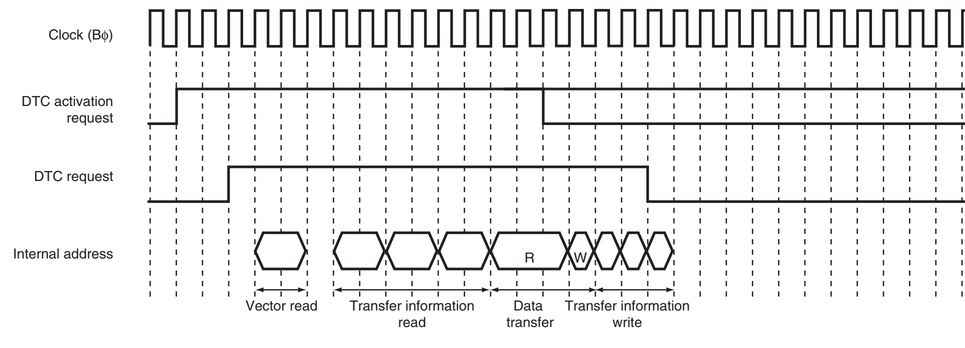
**Figure 8.10 Example of DTC Operation Timing: Normal Transfer Mode or Repeat Transfer Mode (Activated by On-Chip Peripheral Module; Iϕ: Bϕ: Pϕ =1: 1/2: 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)**



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)

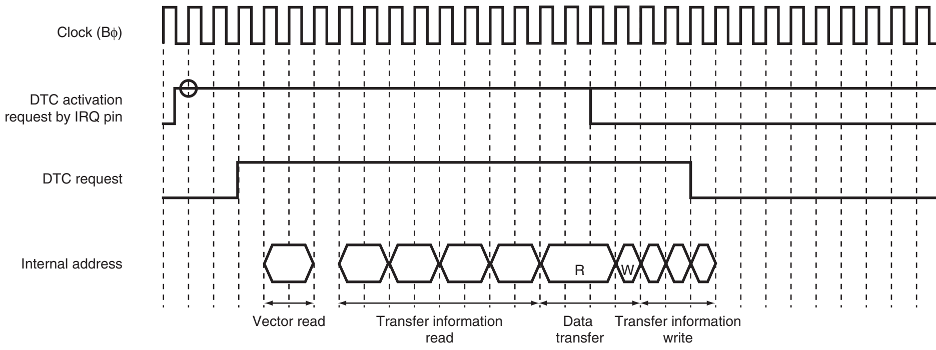


Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 8.13 Example of DTC Operation Timing: 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)

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

**Figure 8.14 Example of DTC Operation Timing: Normal 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)**


Clock (Bϕ)
DTC activation request by IRQ pin
DTC request
Internal address
R
W
Vector read
Transfer information read
Data transfer
Transfer information write

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

**Figure 8.15 Example of DTC Operation Timing: Normal 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)**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Repeat | 1 | 0*1 | 4 | 3*4 | 0*1 | 3 | 2*2 | 1*3 | 1 | 1 | 1 |
| Block transfer | 1 | 0*1 | 4 | 3*4 | 0*1 | 3 | 2*2 | 1*3 | 1•P | 1•P | 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

| | | | |
|---|---|---|---|
| Word data read $S_L$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 2Pϕ*3 | 5Bϕ |
| Longword data read $S_L$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 4Pϕ*3 | 9Bϕ |
| Byte data write $S_M$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 2Pϕ*3 | 2Bϕ*5 |
| Word data write $S_M$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 2Pϕ*3 | 2Bϕ*5 |
| Longword data write $S_M$ | 1Bϕ to 3Bϕ*1 | 1Bϕ + 4Pϕ*3 | 2Bϕ*5 |
| Internal operation $S_N$ | 1 | | |

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

| | 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 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 internal bus, replace 1Bϕ by 1Pϕ in a slow case.
4. Values are different depending on the BSC register setting. The values in the 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 case where writing is frequently executed, such as block transfer, and when t external bus is in use because the write buffer cannot be used efficiently in su For details on the write buffer, see section 9.5.7 (2), Access in View of LSI Int Master.

not release the bus mastership during transfer information read, single data transfer, or tra
information writeback.
The bus mastership release timing can be specified through the bus function extending re
(BSCEHR). For details see section 9.4.4, Bus Function Extending Register (BSCEHR). T
difference in bus mastership release timing according to the register setting is summarize
8.11. Settings other than settings 1 to 5 are not allowed. The setting must not be changed
DTC is active.

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

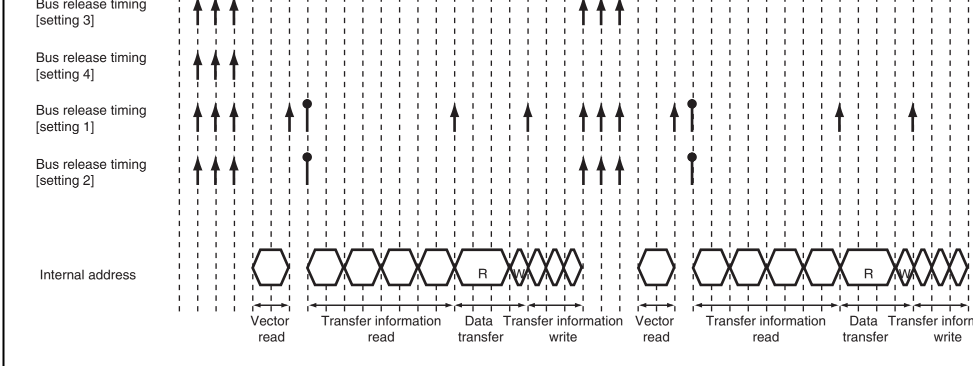
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Setting 4*2 | 0 | 1 | *3 | *3 | 1 | x | x | x | x | O |
| Setting 5 | 1 | 1 | *3 | 1 | 0 | O | x | O | O | O |

Notes: 1. The bus mastership is only released for the external space access request fro 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 between external memory and on-chip peripheral module.

3. Don't care.


Bus release timing [setting 3]
Bus release timing [setting 4]
Bus release timing [setting 1]
Bus release timing [setting 2]
Internal address
R
W
R
W
Vector read
Transfer information read
Data transfer
Transfer information write
Vector read
Transfer information read
Data transfer
Transfer inform
write

[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)**

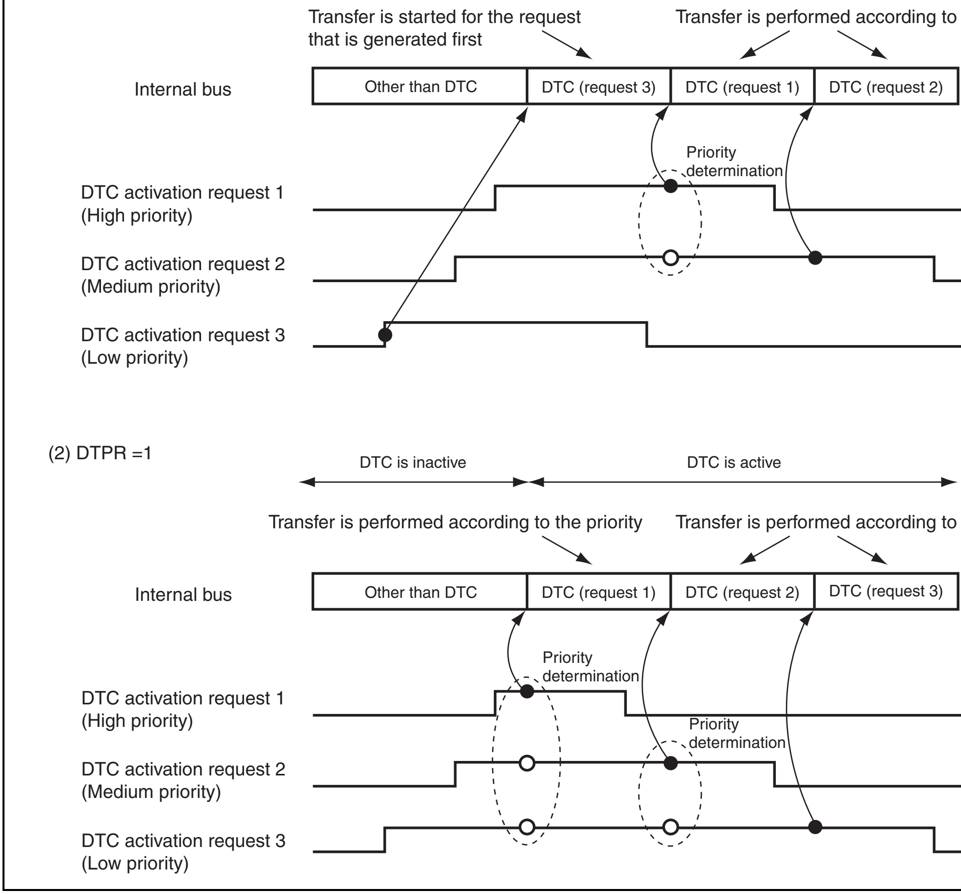
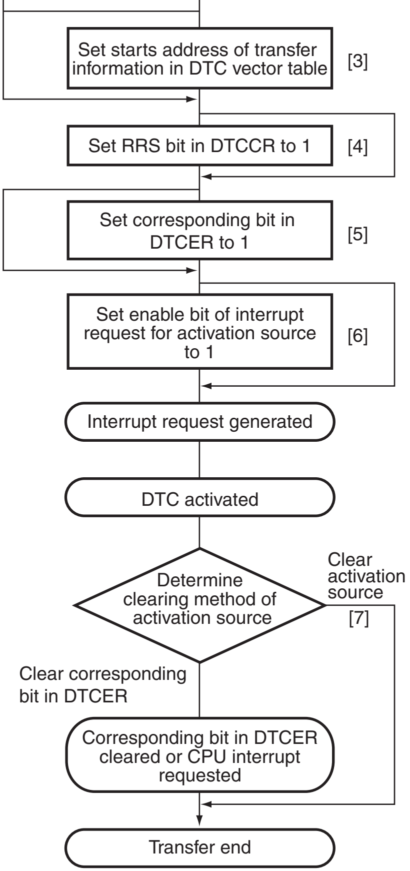


**Figure 8.17 Example of DTC Activation in Accordance with Priority**


Set starts address of transfer information in DTC vector table
[3]
Set RRS bit in DTCCR to 1
[4]
Set corresponding bit in DTCER to 1
[5]
Set enable bit of interrupt request for activation source to 1
[6]
Interrupt request generated
DTC activated
Determine clearing method of activation source
Clear activation source
[7]
Clear corresponding bit in DTCER
Corresponding bit in DTCER cleared or CPU interrupt requested
Transfer end

on location of transfer information, see section 8.4, Loc
Transfer Information and DTC Vector Table.

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

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

[5] Set the bit in DTCER corresponding to the DTC activati
interrupt source to 1. For the correspondence of interru
DTCER, refer to table 8.2. The bit in DTCER may be se
the second or later transfer. In this case, setting the bit
needed.

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

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

**Figure 8.18 Activation of DTC by Interrupt**

2. Set the start address of the transfer information for an RXI interrupt at the DTC vect
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 th end (RXI) interrupt. Since the generation of a receive error during the SCI reception 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 se RXI interrupt is generated, and the DTC is activated. The receive data is transferred to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is hel DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. Terminati 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 o 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 to have been set to start at lower address H'0000. Figure 8.19 shows the chain transfer w counter value is 0.

1. For the first transfer, set the normal transfer mode for input data. Set the fixed transf 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 t data transfer in a separate area (in ROM, etc.). For example, if the input buffer is co addresses H'200000 to H'21FFFF, prepare H'21 and H'20.

data transfer is started. Set the upper eight bits of the transfer destination address for t
data transfer to H'20. The lower 16 bits of the transfer destination address of the first c
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 t
no interrupt request is sent to the CPU.

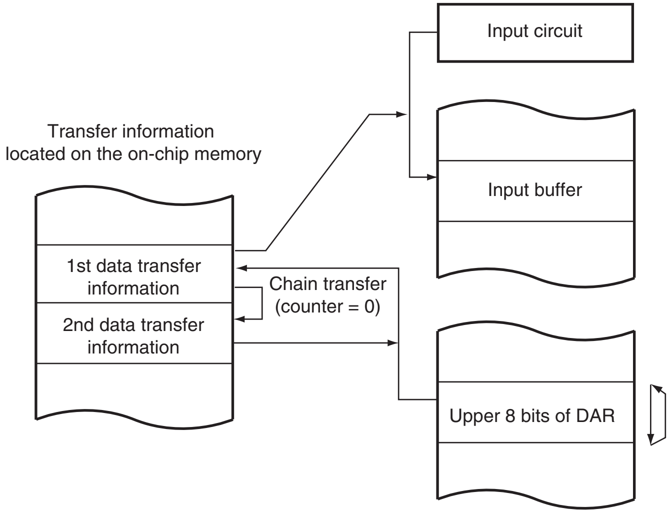


**Figure 8.19 Chain Transfer when Counter = 0**

### 8.9.1 Module Standby Mode Setting

Operation of the DTC can be disabled or enabled using the standby control register. The setting is for operation of the DTC to be disabled. DTC operation is disabled in module mode but register access is available. Module standby mode cannot be set while the DT activated. Before entering software standby mode or module standby mode, all DTCER 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 RA 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 wri 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 w last of the chain of data transfers is executed. SCI and A/D converter interrupt/activatio 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 Ad

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

an interrupt to the CPU (transfer counter = 0 or DISEL = 1), the IRQ signal must be k
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 TEN
  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 handler in the same way as for general interrupt source flags. For details, refer to section 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 activati 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 determinin 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 Progres

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

— 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 same space/different space), read-read (in same space/different space), the first c write access.

2. Normal space interface

— Supports the interface that can directly connect to the SRAM

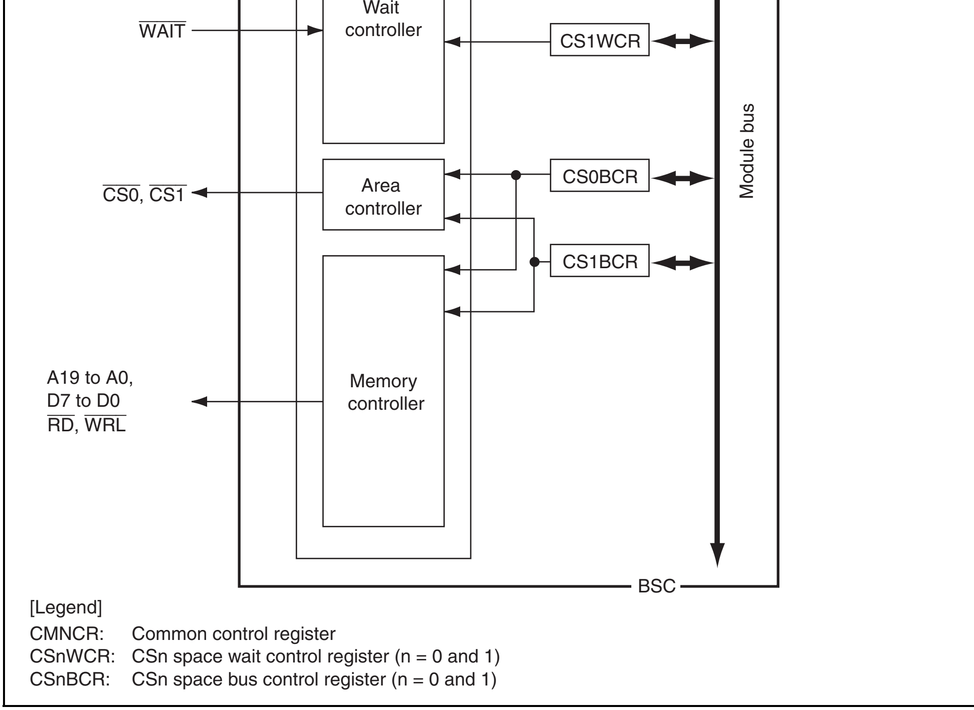

Wait
controller
WAIT
CS1WCR
Module bus
Area
controller
CS0, CS1
CS0BCR
CS1BCR
A19 to A0,
D7 to D0
RD, WRL
Memory
controller
BSC
[Legend]
CMNCR: Common control register
CSnWCR: CSn space wait control register (n = 0 and 1)
CSnBCR: CSn space bus control register (n = 0 and 1)

**Figure 9.1 Block Diagram of BSC**

|  |  |  |
|---|---|---|
| $\overline{\text{WRL}}$ | Output | Indicates byte write through D7 to D0. |
| $\overline{\text{WAIT}}$ | Input | External wait input |
| $\overline{\text{BREQ}}$ | Input | Bus request input |
| $\overline{\text{BACK}}$ | 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, an
outputs chip select signals ($\overline{\text{CS0}}$ and $\overline{\text{CS1}}$) for each of them. $\overline{\text{CS0}}$ is asserted during area

### 9.3.2 Address Map

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

| | | | | |
|---|---|---|---|---|
| H'FFFFFFFF | modules | | | |

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

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

| Address | Area | Memory Type | Capacity | Bus Widt |
|---|---|---|---|---|
| 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 bi |

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

| | | | | |
|---|---|---|---|---|
| H'03FFFFFF | | | | |
| 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 b |

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

| | | | |
|---|---|---|---|
| H'FFFF9FFF | | | |
| 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 bi |

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

| | | | | |
|---|---|---|---|---|
| H'03FFFFFF | | | | |
| 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 b |

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

|  |  |  |  |
|---|---|---|---|
| H'FFFF7FFF |  |  |  |
| 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 bi |

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

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 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, |

### 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

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

| | | | | |
|---|---|---|---|---|
| 4 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value sh always be 1. |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu always be 0. |
| 1 | HIZMEM | 0 | R/W | High-Z Memory Control<br>Specifies the pin state in software standby moc A19 to A0, $\overline{\text{CSn}}$, $\overline{\text{WRL}}$, and $\overline{\text{RD}}$. While the bus released, these pins are in high-impedance sta 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 sh always be 0. |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0* | 1 | 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 |

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 va always be 0. |
| 29, 28 | IWW[1:0] | 11 | R/W | Specification for Idle Cycles between Write-R Write Cycles<br>Specify the number of idle cycles to be insert access to memory that is connected to the sp target cycles are write-read cycles and 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 s always be 0. |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | 11: 4 idle cycles inserted |
| 24 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value sh<br>always be 0. |
| 23, 22 | IWRWS[1:0] | 11 | R/W | Specification for Idle Cycles between Read-Wr<br>Cycles in the Same Space<br>Specify the number of idle cycles to be inserted<br>access to memory that is connected to the spa<br>target cycles are continuous read-write cycles<br>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 sh<br>always be 0. |
| 20, 19 | IWRRD[1:0] | 11 | R/W | Specification for Idle Cycles between Read-Re<br>Cycles in Different Spaces<br>Specify the number of idle cycles to be inserted<br>access to memory that is connected to the spa<br>target cycles are continuous read-read cycles i<br>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 |

| | | | | |
|---|---|---|---|---|
| | | | | 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 va 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 chip ROM is enabled, write B'01 to specify the width as 8-bit before accessing the CSn spac<br>Note: When the on-chip ROM is disabled, th width of area 0 is 8 bits regardless of t BSZ[1:0] bit setting in CS0BCR. |
| 8 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va always be 0. |

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

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 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 | F |

| 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 valu 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<br>000: The same cycles as WR3 to WR0 settings 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 valu always be 0. |

| | | | | |
|---|---|---|---|---|
| 10 to 7 | WR[3:0] | 1010 | R/W | Number of Read Access Wait Cycles<br>Specify the number of wait cycles required for<br>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 inpu<br>The specification by this bit is valid even wher<br>number of access wait cycles is 0.<br>0: External wait input is valid<br>1: External wait input is ignored |

10: 2.5 cycles

11: 3.5 cycles

### 9.4.4 Bus Function Extending Register (BSCEHR)

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

Setting the CSSTP2 bit can improve the transfer performance of the DTC transfer when t DTLOCK bit is 0. Furthermore, setting the CSSTP3 bit selects whether or not access to th 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 s 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 selects whether or not this priority order is valid or invalid when multiple sources issue ac 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | DTLOCK | CSSTP1 | - | CSSTP2 | DTBST | DTSA | CSSTP3 | DTPR | - | - | - | - | - | - |
| Initial value: | 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 |

| | | | | |
|---|---|---|---|---|
| 14 | CSSTP1 | 0 | R/W | Select Bus Release on NOP Cycle Generation<br>Specifies whether or not the bus is released in<br>to requests from the CPU for external space ac<br>generation of the NOP cycle that follows readir<br>vector address.<br>If, however, the CSSTP2 bit is 1, bus mastersh<br>retained until all transfer is complete, regardles<br>setting of this bit.<br>0: The bus is released on generation of the NC<br>the DTC.<br>1: The bus is not released on generation of the<br>cycle by the DTC. |
| 13 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value sh<br>always be 0. |
| 12 | CSSTP2 | 0 | R/W | Select Bus Release during DTC Transfer<br>This setting applies to DTC transfer when the D<br>bit is 0. The value specifies whether the bus ma<br>is or is not to be released after each round of tr<br>response to a request from the CPU for access<br>external space.<br>0: When the DTLOCK and CSSTP1 bits are 0,<br>released on generation of the NOP cycle aft<br>of the vector address. When the DTLOCK bi<br>the CSSTP1 bit is 1, the bus is released afte<br>round of data transfer.<br>1: Only release the bus mastership after all dat<br>is complete. |

| | | | | |
|---|---|---|---|---|
| | | | | activation sources.<br>Notes: When this bit is set to 1, the following res<br>apply.<br>1. Clock setting with the frequency contro<br>(FRQCR) must be Iϕ: Bϕ: Pϕ: MIϕ: MP<br>4: 4: 4, 4: 2: 2: 2: 2, or 2: 1: 1: 1: 1<br>2. The vector information must be in on-c<br>or on-chip RAM.<br>3. The transfer information must be in on<br>RAM.<br>4. Transfer must be between the on-chip<br>and an on-chip peripheral module or b<br>external memory and an on-chip perip<br>module. |
| 10 | DTSA | 0 | R/W | DTC Short Address Mode<br>In this mode, the information that specifies a DT<br>transfer takes up only 3 longwords.<br>0: Transfer information is read out as 4 longword<br>transfer information is arranged as shown in f<br>(normal address mode).<br>1: Transfer information is read out as 3 longword<br>transfer information is arranged as shown in f<br>(short address mode).<br>Note: Transfer in short address mode is only ava<br>between on-chip peripheral modules and o<br>RAM, because the higher-order 8 bits of th<br>and DAR are considered to be all 1. |

|  |  |  |  |  |
|---|---|---|---|---|
| 8 | DTPR | 0 | R/W | Application of Priority in DTC Activation<br>When multiple DTC activation requests are ger before the DTC is activated, specify whether tr starts from the first request to have been gener in accord with the priority order for DTC activat requests.<br>However, when multiple DTC activation reques been issued while the DTC is active, the next t be triggered will be that with the highest DTC a priority.<br>0: Start transfer in response to the first request been generated.<br>1: Start transfer in accord with DTC activation r priority.<br>Notes: When this bit is set to 1, the following re apply.<br>1. The vector information must be in on- or on-chip RAM.<br>2. The transfer information must be in o RAM.<br>3. Skipping of transfer information readi always disabled. |
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu always be 0. |

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

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

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

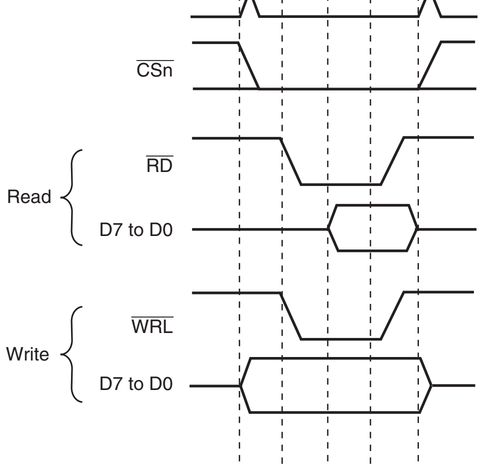


**Figure 9.2 Normal Space Basic Access Timing (Access Wait 0)**

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

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

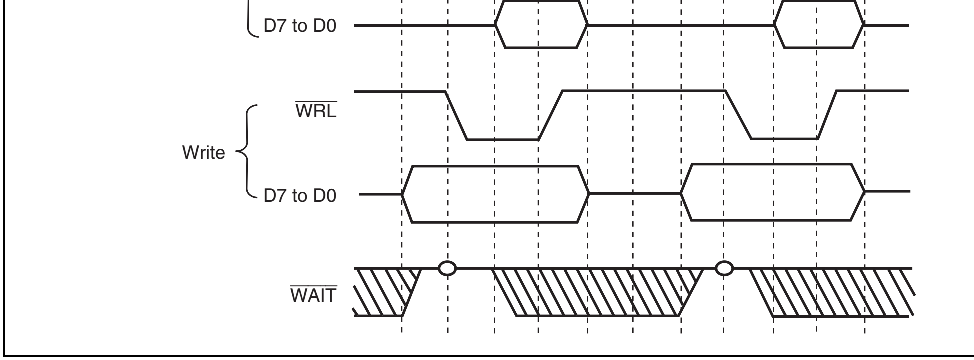


**Figure 9.3 Continuous Access for Normal Space 1 Bus Width = 8 Bits, Word Access, WM Bit in CSnWCR = 0 (Access Wait = 0, Cycle Wait = 0)**

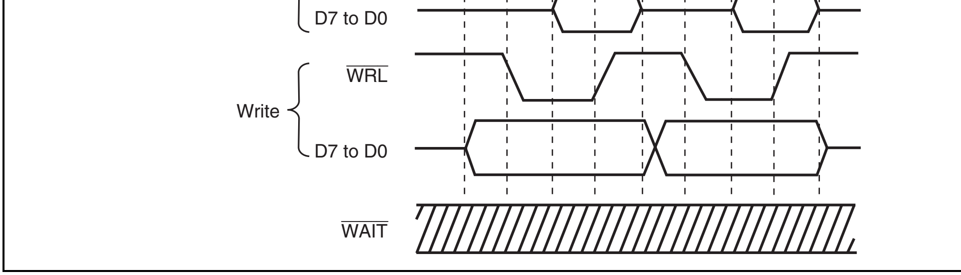


**Figure 9.4 Continuous Access for Normal Space 2 Bus Width = 8 Bits, Word Access, WM Bit in CSnWCR = 1 (Access Wait = 0, Cycle Wait = 0)**

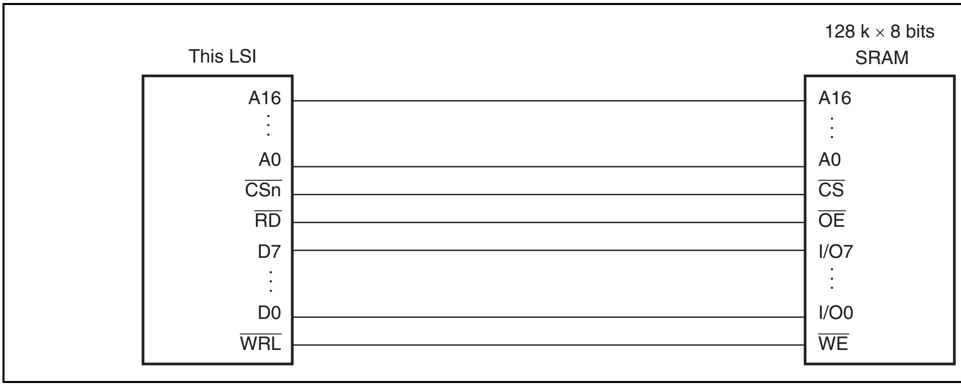


**Figure 9.5 Example of 8-Bit Data-Width SRAM Connection**

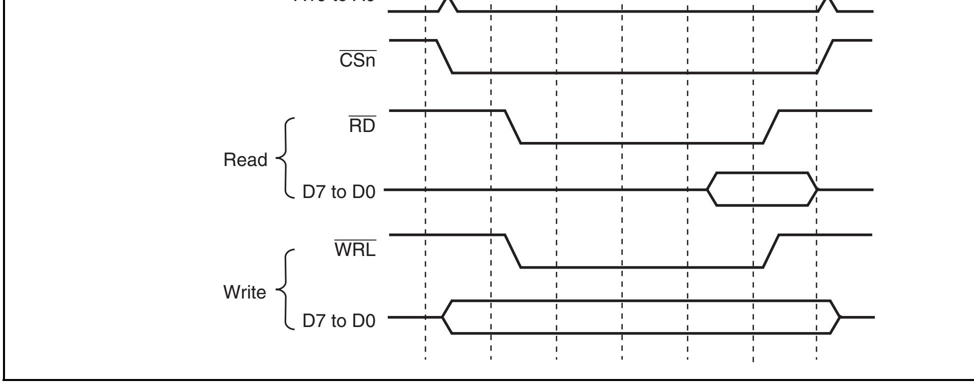


**Figure 9.6 Wait Timing for Normal Space Access (Software Wait Only)**

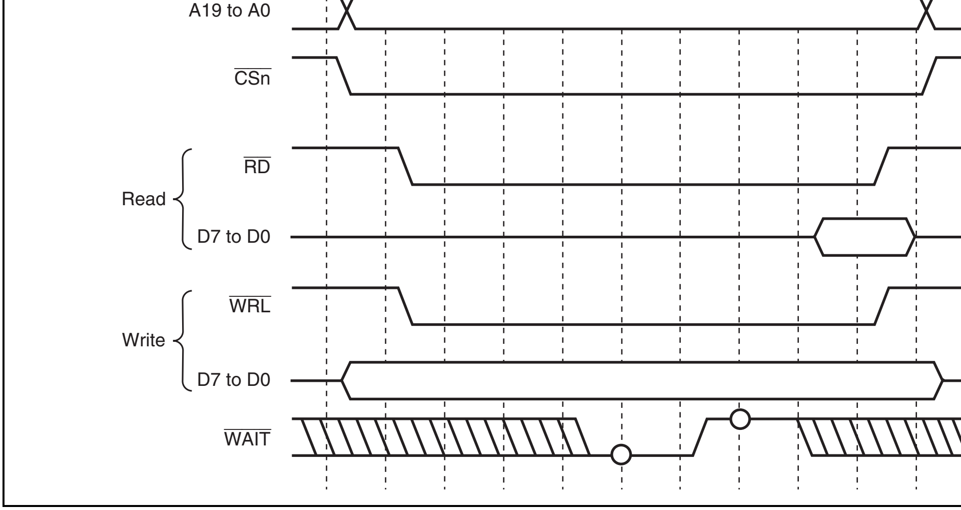


**Figure 9.7 Wait State Timing for Normal Space Access (Wait State Insertion Using $\overline{\text{WAIT}}$ Signal)**

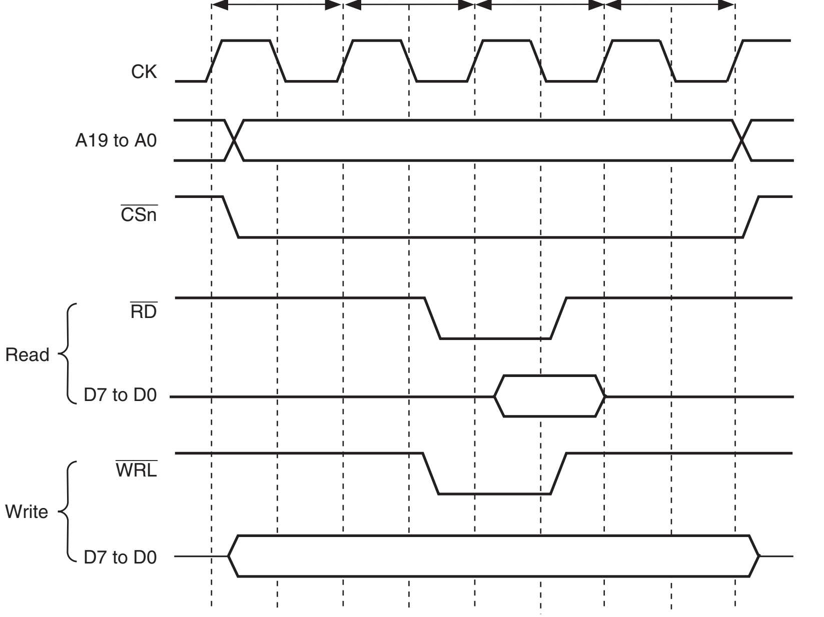


**Figure 9.8 $\overline{\text{CSn}}$ Assert Period Extension**

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 inserted to reserve the minimum pulse width for a multiplexed pin ($\overline{\text{WRL}}$), and an interf 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 bu cycle or a read cycle.
   B. Insert one idle cycle to transfer the read data to the internal bus when a read cycl external bus terminates.

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

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

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 2/2/2/2 | 2/2/2/2 | 2/2/2/2 | 2/2/2/2 | 3/3/3/4 | 2/ |
| 0 | 2 | 2/2/2/2 | 2/2/2/2 | 2/2/2/2 | 2/2/2/2 | 3/3/3/4 | 2/ |
| 1 | 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/ |

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

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

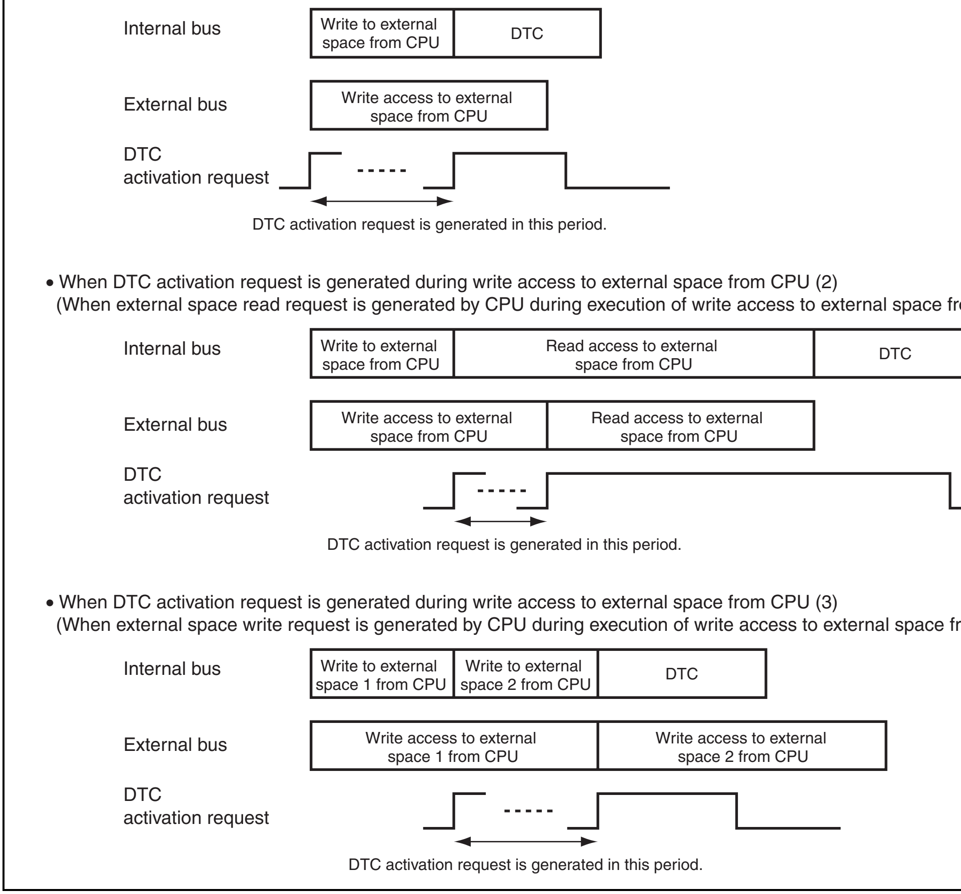
| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 2 | 2 |
| 0 | 2 | 2 | 2 | 2 | 2 |
| 1 | 4 | 4 | 4 | 4 | 4 |
| 0 | 4 | 4 | 4 | 4 | 4 |

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

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

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

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


Internal bus
Write to external space from CPU
DTC
External bus
Write access to external space from CPU
DTC activation request
DTC activation request is generated in this period.

- When DTC activation request is generated during write access to external space from CPU (2) (When external space read request is generated by CPU during execution of write access to external space fr

Internal bus
Write to external space from CPU
Read access to external space from CPU
DTC
External bus
Write access to external space from CPU
Read access to external space from CPU
DTC activation request
DTC activation request is generated in this period.

- When DTC activation request is generated during write access to external space from CPU (3) (When external space write request is generated by CPU during execution of write access to external space fr

Internal bus
Write to external space 1 from CPU
Write to external space 2 from CPU
DTC
External bus
Write access to external space 1 from CPU
Write access to external space 2 from CPU
DTC activation request
DTC activation request is generated in this period.

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

connected to these control signals.

Bus mastership is transferred to the external device at the boundary of bus cycles. Namel
mastership is released immediately after receiving a bus request when a bus cycle is not b
performed. The release of bus mastership is delayed until the bus cycle is complete when
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 betw
access cycles. Therefore, it cannot be immediately determined whether or not bus masters
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 ove
the slave has many tasks, multiple bus cycles should be executed in a bus mastership acq
Reducing the cycles required for master to slave bus mastership transitions streamlines th
design.

The LSI has the bus mastership until a bus request is received from the external device. U
acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LSI
the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the
acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the slave has re
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 dev
unless an external device is accessed. When an external device is accessed, the LSI enters
of waiting for bus mastership to be returned.

While the bus is released, sleep mode, software standby mode, and deep software standby
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 acknowled
is asserted 0.5 cycles after the above high impedance state, synchronized with the falling

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

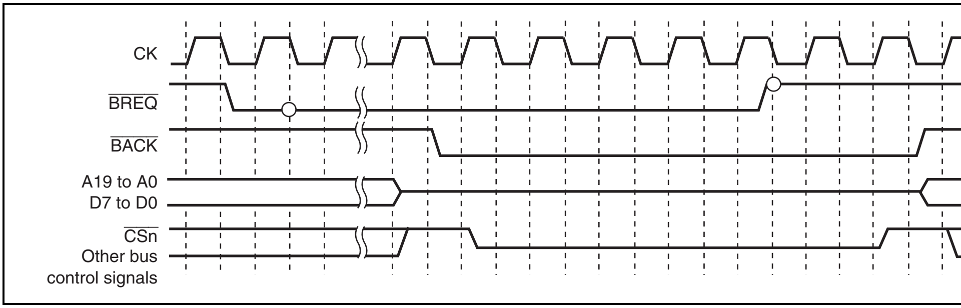


**Figure 9.10 Bus Arbitration Timing**

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

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 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 bus. Thus, the DTC can be activated without bus arbitration with the CPU while the CPU accessing an on-chip memory.

Since the bus state controller (BSC) incorporates a one-stage write buffer, the BSC can ex access via the I bus before the previous external bus cycle is completed in a write cycle. I chip peripheral module is read or written after the external low-speed memory is written, chip peripheral module can be accessed before the completion of the external low-speed write cycle.

In read cycles, the CPU is placed in the wait state until read operation has been complete continue the process after the data write to the device has been completed, perform a dum to the same address to check for completion of the write before the next process to be exe

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 write buffer is operating, do not modify BSC registers immediately after a write access. I register need to be modified immediately after a write access, execute dummy read to co completion of the write access, then modify the BSC register.

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 e
are input and output in synchronization with rising edges of the corresponding clock sig
bus, I bus, and peripheral bus are synchronized with the Iϕ, Bϕ, and Pϕ clock, respective
9.11 shows an example of the timing of write access to a register in 2Pϕ cycle access wi
connected peripheral bus width of 16 bits when Iϕ:Bϕ:Pϕ = 4:2:2. In access to the on-ch
peripheral I/O registers, the CPU requires three cycles of Iϕ for preparation of data trans
bus after the data has been output to the L bus. After these three cycles, data can be trans
the I bus in synchronization with rising edges of Bϕ. However, as there are two Iϕ clock
a single Bϕ clock cycle when Iϕ: Bϕ = 4:2, transfer of data from the L bus to the I bus ta

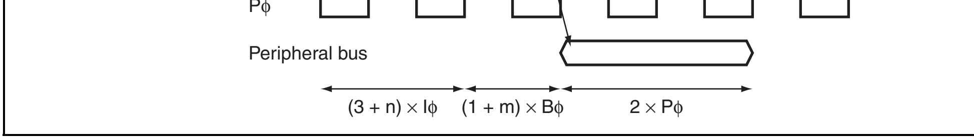


**Figure 9.11 Timing of Write Access to On-Chip Peripheral I/O Registers When Iϕ:Bϕ:Pϕ = 4:2:2**

Figure 9.12 shows an example of timing of read access to the peripheral bus when Iϕ:Bϕ: 4:2:1. Transfer from the L bus to the peripheral bus is performed in the same way as for v the case of reading, however, values output onto the peripheral bus need to be transferred CPU. Although transfers from the peripheral bus to the I bus and from the I bus to the L b performed in synchronization with the rising edge of the respective bus clocks, a period c is actually required because Iϕ ≥ Bϕ ≥ Pϕ. In the case shown in the figure, where n = 0 an the time required for access is 3 × Iϕ + 2 × Bϕ + 2 × Pϕ + 2 × Iϕ.

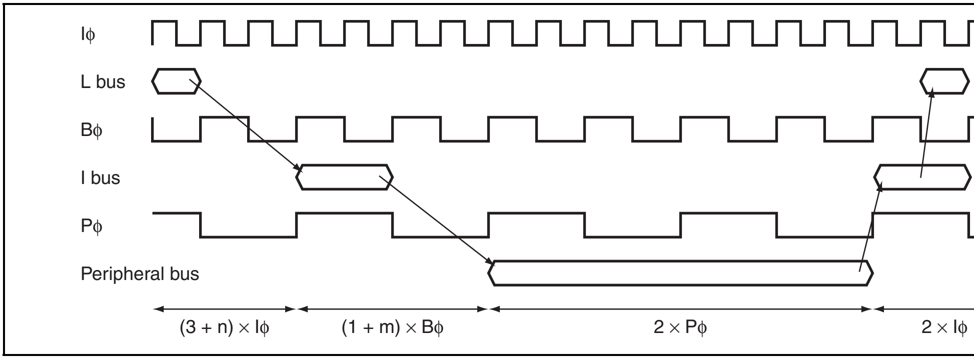


**Figure 9.12 Timing of Read Access to On-Chip Peripheral I/O Registers When Iϕ:Bϕ:Pϕ = 4:2:1**

|  |  |  |  |
|---|---|---|---|
| 8 bits | Byte | Write | (1 + n) × Iϕ + (3 + m) × Bϕ |
|  |  | Read | (1 + n) × Iϕ + (3 + m) × Bϕ + 1 × Iϕ |
|  | Word | Write | (1 + n) × Iϕ + (3 + m) × Bϕ + 1 × (2 + o) × B |
|  |  | Read | (1 + n) × Iϕ + (3 + m) × Bϕ + 1 × (2 + o) × B |
|  | Longword | Write | (1 + n) × Iϕ + (3 + m) × Bϕ + 3 × (2 + o) × B |
|  |  | Read | (1 + n) × Iϕ + (3 + m) × Bϕ + 3 × (2 + o) × B |

Notes: n: When Iϕ:Bϕ = 8:1, n = 0 to 7
When Iϕ:Bϕ = 4:1, n = 0 to 3
When Iϕ:Bϕ = 3:1, n = 0 to 2
When Iϕ:Bϕ = 2:1, n = 0 to 1
When Iϕ:Bϕ = 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. Da
bus are input and output in synchronization with rising edges of the corresponding clock
The L bus and I bus are synchronized with the Iϕ and Bϕ clocks, respectively. Figure 9.
an example of the timing of write access to a word of data over the external bus, with a b
of 8 bits, when Iϕ:Bϕ = 2:1. Once the CPU has output the data to the L bus, data are tran
the I bus in synchronization with rising edges of Bϕ. There are two Iϕ clock cycles in a s
clock cycle when Iϕ: Bϕ = 2:1. Thus, when Iϕ: Bϕ = 2:1, data transfer from the L bus to
takes (1 + n) × Iϕ (n = 0 to 1) (2 × Iϕ is indicated in figure 9.13). The relation between th
of data output to the L bus and the rising edge of Bϕ depends on the state of program ex
Data output to the I bus are transferred to the external bus after one cycle of Bϕ. Externa
each data takes at least two cycles, and this can be prolonged by the BSC register setting
o in the formulae for number of access cycles). In the case shown in figure 9.13, since n
0, and o = 0, access takes 2 × Iϕ + 3 × Bϕ + 2 × Bϕ.

**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
than or equal to the data width and Iϕ:Bϕ = 4:1. Transfer from the L bus to the external bu
performed in the same way as for write access. In the case of reading, however, values ou
the external bus must be transferred to the CPU. Transfers from the external bus to the I b
from the I bus to the L bus are again performed in synchronization with rising edges of th
respective bus clocks. In the actual operation, transfer from the external bus to the L bus t
ϕ period. In the case shown in the figure, where n = 2 and m = 0, access takes 3 × Iϕ + 3 ×
× Iϕ.

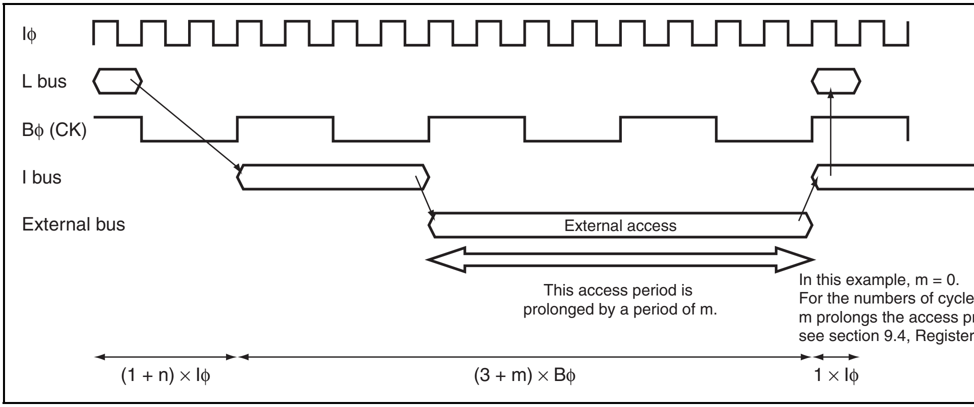


**Figure 9.14 Timing of Read Access with Condition Iϕ:Bϕ = 4:1 and External Bus Width ≥ Data Width**

For access by the DTC, the access cycles are obtained by subtracting the cycles of Iϕ requ
L-bus access from the access cycles required for access by the CPU.

- 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

- 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, positive and negative phases of reset PWM output by interlocking operation of chan 4, is possible.
- AC synchronous motor (brushless DC motor) drive mode using complementary PW and reset PWM output is settable by interlocking operation of channels 0, 3, and 4, a 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 valu converter start triggers can be skipped.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | TGRB_0<br>TGRE_0 | TGRB_1 | TGRB_2 | TGRB_3 | TGRB_4 | TG<br>TG |
| 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 | Inp<br>TIC<br>TIC<br>TIC |
| 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 | TG con mat inpu |
| 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 | | √ | — | — | √ | √ | — |

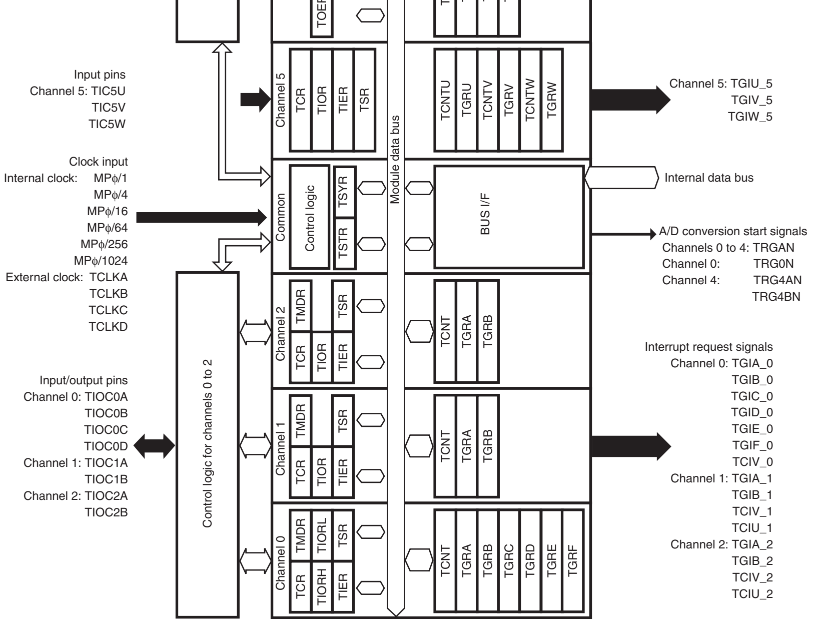
| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | and TCNT overflow or underflow | ca |
| A/D converter start trigger | TGRA_0 compare match or input capture<br>TGRE_0 compare match | TGRA_1 compare match or input capture | TGRA_2 compare match or input capture | TGRA_3 compare match or input capture | TGRA_4 compare match or input capture<br>TCNT_4 underflow (trough) in complementary PWM mode | — |

| | | | | | | |
|---|---|---|---|---|---|---|
| | input capture 0C<br>• Compare match or input capture 0D<br>• Compare match 0E<br>• Compare match 0F<br>• Overflow | • Underflow | • Underflow | input capture 3C<br>• Compare match or input capture 3D<br>• Overflow | input capture 4C<br>• Compare match or input capture 4D<br>• Overflow or underflow | in ca |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | 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 |



[Legend]

TSTR: Timer start register
TSYR: Timer synchronous register
TCR: Timer control register
TMDR: Timer mode register
TIOR: Timer I/O control register
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIER: Timer interrupt enable register
TGCR: Timer gate control register
TOER: Timer output master enable register
TOCR: Timer output control register
TSR: Timer status register
TCNT: Timer counter
TCNTS: Timer subcounter

TCDR: Timer cycle data register
TCBR: Timer cycle buffer register
TDDR: Timer dead time data register
TGRA: Timer general register A
TGRB: Timer general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TGRU: Timer general register U
TGRV: Timer general register V
TGRW: Timer general register W

**Figure 10.1 Block Diagram of MTU2**

| | | | |
|---|---|---|---|
| | 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 |
| | TIOC0B | I/O | TGRB_0 input capture input/output compare output/PWM |
| | TIOC0C | I/O | TGRC_0 input capture input/output compare output/PWM |
| | TIOC0D | I/O | TGRD_0 input capture input/output compare output/PWM |
| 1 | TIOC1A | I/O | TGRA_1 input capture input/output compare output/PWM |
| | TIOC1B | I/O | TGRB_1 input capture input/output compare output/PWM |
| 2 | TIOC2A | I/O | TGRA_2 input capture input/output compare output/PWM |
| | TIOC2B | I/O | TGRB_2 input capture input/output compare output/PWM |
| 3 | TIOC3A | I/O | TGRA_3 input capture input/output compare output/PWM |
| | TIOC3B | I/O | TGRB_3 input capture input/output compare output/PWM |
| | TIOC3C | I/O | TGRC_3 input capture input/output compare output/PWM |
| | TIOC3D | I/O | TGRD_3 input capture input/output compare output/PWM |
| 4 | TIOC4A | I/O | TGRA_4 input capture input/output compare output/PWM |
| | TIOC4B | I/O | TGRB_4 input capture input/output compare output/PWM |
| | TIOC4C | I/O | TGRC_4 input capture input/output compare output/PWM |
| | TIOC4D | I/O | TGRD_4 input capture input/output compare output/PWM |
| 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 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Timer control register_3 | TCR_3 | R/W | H'00 | H'FFFFC200 | 8, 1 |
| 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, 1 |
| 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, 1 |
| 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, 1 |
| 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, 1 |
| 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, 1 |
| 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, |
| Timer counter_4 | TCNT_4 | R/W | H'0000 | H'FFFFC212 | 16 |
| Timer cycle data register | TCDR | R/W | H'FFFF | H'FFFFC214 | 16, |
| 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, |
| 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, |
| Timer general register B_4 | TGRB_4 | R/W | H'FFFF | H'FFFFC21E | 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, |
| 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, |
| 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 |
| 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 |
| Timer A/D converter start request cycle set buffer register B_4 | TADCOBRB_4 | R/W | H'FFFF | H'FFFFC24A | 16 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Timer control register_0 | TCR_0 | R/W | H'00 | H'FFFFC300 | 8, 1 |
| 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, 1 |
| 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, 1 |
| 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, |
| 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, |
| 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, |
| 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, 1 |
| 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, 1 |
| 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, 1 |
| Timer status register_1 | TSR_1 | R/W | H'C0 | H'FFFFC385 | 8 |

| | | | | | |
|---|---|---|---|---|---|
| Timer interrupt enable register_2 | TIER_2 | R/W | H'00 | H'FFFFC404 | 8, |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |

| 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 so 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 th clock is counted using both edges, the input cl period is halved (e.g. MPϕ/4 both edges = MP edge). If phase counting mode is used on char and 2, this setting is ignored and the phase co mode setting has priority. Internal clock edge s is valid when the input clock is MPϕ/4 or slowe MPϕ/1, or the overflow/underflow of another ch selected for the input clock, although values ca written, counter operation compiles with the ini<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 source can be selected independently for each See tables 10.6 to 10.10 for details. |

[Legend]

x: Don't care

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | synchronous operation* |
|  | 1 | 0 | 0 | TCNT clearing disabled |
|  |  |  | 1 | TCNT cleared by TGRC compare mat capture*2 |
|  |  | 1 | 0 | TCNT cleared by TGRD compare mat capture*2 |
|  |  |  | 1 | TCNT cleared by counter clearing for a channel performing synchronous clear 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 beca buffer register setting has priority, and compare match/input capture does not

**Table 10.5 CCLR0 to CCLR2 (Channels 1 and 2)**

| Channel | Bit 7<br>Reserved*2 | Bit 6<br>CCLR1 | Bit 5<br>CCLR0 | Description |
|---|---|---|---|---|
| 1, 2 | 0 | 0 | 0 | TCNT clearing disabled |
|  |  |  | 1 | TCNT cleared by TGRA compare ma capture |
|  |  | 1 | 0 | TCNT cleared by TGRB compare ma capture |
|  |  |  | 1 | TCNT cleared by counter clearing for channel performing synchronous clea 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 mo

| | | | | |
|---|---|---|---|---|
| | | 1 | 0 | External clock: counts on TCLKC pin in |
| | | | 1 | External clock: counts on TCLKD pin in |

**Table 10.7 TPSC0 to TPSC2 (Channel 1)**

| Channel | Bit 2<br>TPSC2 | Bit 1<br>TPSC1 | Bit 0<br>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 in |
| | | | 1 | External clock: counts on TCLKB pin in |
| | | 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.

| | | | | |
|---|---|---|---|---|
| | | 1 | 0 | External clock: counts on TCLKC pin i |
| | | | 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<br>TPSC2 | Bit 1<br>TPSC1 | Bit 0<br>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 i |
| | | | 1 | External clock: counts on TCLKB pin i |

### 10.3.2 Timer Mode Register (TMDR)

The TMDR registers are 8-bit readable/writable registers that are used to set the operating each channel. The MTU2 has five TMDR registers, one each for channels 0 to 4. TMDR 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 sh always be 0. |
| 6 | BFE | 0 | R/W | Buffer Operation E<br>Specifies whether TGRE_0 and TGRF_0 are t in the normal way or to be used together for bu operation. Compare match with TGRF occurs when TGRF is used as a buffer register.<br>In channels 1 to 4, this bit is reserved. It is alwa 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 buf operation |

| | | | | |
|---|---|---|---|---|
| | | | | interrupt enable register 3/4 (TIER_3/4) shoul<br>cleared to 0.<br>In channels 1 and 2, which have no TGRD, bi<br>reserved. It is always read as 0 and cannot b<br>0: TGRB and TGRD operate normally<br>1: TGRB and TGRD used together for buffer |
| 4 | BFA | 0 | R/W | Buffer Operation A<br>Specifies whether TGRA is to operate in the n<br>way, or TGRA and TGRC are to be used toge<br>buffer operation. When TGRC is used as a bu<br>register, TGRC input capture/output compare<br>take place in modes other than complementa<br>mode, but compare match with TGRC occurs<br>complementary PWM mode. Since the TGFC<br>be set if a compare match occurs on channel<br>Tb interval in complementary PWM mode, the<br>bit in timer interrupt enable register 4 (TIER_4<br>be cleared to 0.<br>In channels 1 and 2, which have no TGRC, bi<br>reserved. It is always read as 0 and cannot b<br>0: TGRA and TGRC operate normally<br>1: TGRA and TGRC used together for buffer |
| 3 to 0 | MD[3:0] | 0000 | R/W | Modes 0 to 3<br>These bits are used to set the timer operating<br>See table 10.11 for details. |

| | | | | |
|---|---|---|---|---|
| | | 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)* |
| | | | 1 | Complementary PWM mode 2 (transmit at crest anc 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 s channel 3. When channel 3 is set to reset synchronous PWM mode or comple PWM mode, the channel 4 settings become ineffective and automatically conf channel 3 settings. However, do not set channel 4 to reset synchronous PWM complementary PWM mode. Reset synchronous PWM mode and complement mode cannot be set for channels 0, 1, and 2.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the
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: 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 valu 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. |

| | | | | | |
|---|---|---|---|---|---|
| | | | 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/coun<br>Input capture at TCNT_1 count-up/coun |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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<br>Input capture at TCNT_1 count-up/count |

[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 regis setting is invalid and input capture/output compare is not generated.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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 match/input capture |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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 regis setting is invalid and input capture/output compare is not generated.

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  | 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.

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  | 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 regis setting is invalid and input capture/output compare is not generated.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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/coun<br>Input capture at TCNT_1 count-up/cour |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  | 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<br>Input capture at TCNT_1 count-up/count |

[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 regis setting is invalid and input capture/output compare is not generated.

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  | 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 compare match/input capture |

[Legend]

x: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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.

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  | 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 regis setting is invalid and input capture/output compare is not generated.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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.

| | | | | | |
|---|---|---|---|---|---|
| | | | 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 regis setting is invalid and input capture/output compare is not generated.

| | | | | | | |
|---|---|---|---|---|---|---|
| | 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 in<br>Capture at trough of complementary PWM mo |
| | | | 1 | 0 | | Measurement of low pulse width of external in<br>Capture at crest of complementary PWM mod |
| | | | | 1 | | Measurement of low pulse width of external in<br>Capture at crest and trough of complementary mode |
| | | 1 | 0 | 0 | | Setting prohibited |
| | | | | 1 | | Measurement of high pulse width of external i<br>Capture at trough of complementary PWM mo |
| | | | 1 | 0 | | Measurement of high pulse width of external i<br>Capture at crest of complementary PWM mod |
| | | | | 1 | | Measurement of high pulse width of external i<br>Capture at crest and trough of complementary mode |

[Legend]

x: Don't care

| | | | | |
|---|---|---|---|---|
| 7 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu always be 0. |
| 2 | CMPCLR5U | 0 | R/W | TCNT Compare Clear 5U<br>Enables or disables requests to clear TCNTU_ TCNTU_5 and TGRU_5 compare match or inp capture.<br>0: Disables TCNTU_5 to be cleared to H'0000 TCNTU_5 and TGRU_5 compare match or i capture<br>1: Enables TCNTU_5 to be cleared to H'0000 a TCNTU_5 and TGRU_5 compare match or i capture |
| 1 | CMPCLR5V | 0 | R/W | TCNT Compare Clear 5V<br>Enables or disables requests to clear TCNTV_ TCNTV_5 and TGRV_5 compare match or inp capture.<br>0: Disables TCNTV_5 to be cleared to H'0000 TCNTV_5 and TGRV_5 compare match or i capture<br>1: Enables TCNTV_5 to be cleared to H'0000 a TCNTV_5 and TGRV_5 compare match or i capture |

### 10.3.5 Timer Interrupt Enable Register (TIER)

The TIER registers are 8-bit readable/writable registers that control enabling or disablin
interrupt requests for each channel. The MTU2 has seven TIER registers, two for chann
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 conver<br>requests by TGRA input capture/compare ma<br>0: A/D converter start request generation disa<br>1: A/D converter start request generation ena |

| | | | | |
|---|---|---|---|---|
| | | | | underflow (trough) enabled |
| 5 | TCIEU | 0 | R/W | Underflow Interrupt Enable<br>Enables or disables interrupt requests (TCIU) b<br>TCFU flag when the TCFU flag in TSR is set to<br>channels 1 and 2.<br>In channels 0, 3, and 4, bit 5 is reserved. It is a<br>read as 0 and the write value should always be<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) b<br>TCFV flag when the TCFV flag in TSR is set to<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) b<br>TGFD bit when the TGFD bit in TSR is set to 1<br>channels 0, 3, and 4.<br>In channels 1 and 2, bit 3 is reserved. It is alwa<br>as 0 and the write value should always be 0.<br>0: Interrupt requests (TGID) by TGFD bit disab<br>1: Interrupt requests (TGID) by TGFD bit enabl |

| | | | | |
|---|---|---|---|---|
| | | | | Enables or disables interrupt requests (TGIB) TGFB bit when the TGFB bit in TSR is set to<br>0: Interrupt requests (TGIB) by TGFB bit disa<br>1: Interrupt requests (TGIB) by TGFB bit enab |
| 0 | TGIEA | 0 | R/W | TGR Interrupt Enable A<br>Enables or disables interrupt requests (TGIA) TGFA bit when the TGFA bit in TSR is set to<br>0: Interrupt requests (TGIA) by TGFA bit disa<br>1: Interrupt requests (TGIA) by TGFA bit enab |

| | | | | |
|---|---|---|---|---|
| | | | | TGRE_0.<br>0: A/D converter start request generation by co<br>match between TCNT_0 and TGRE_0 disab<br>1: A/D converter start request generation by co<br>match between TCNT_0 and TGRE_0 enab |
| 6 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu<br>always be 0. |
| 1 | TGIEF | 0 | R/W | TGR Interrupt Enable F<br>Enables or disables interrupt requests by comp<br>match between TCNT_0 and TGRF_0.<br>0: Interrupt requests (TGIF) by TGFE bit disab<br>1: Interrupt requests (TGIF) by TGFE bit enabl |
| 0 | TGIEE | 0 | R/W | TGR Interrupt Enable E<br>Enables or disables interrupt requests by comp<br>match between TCNT_0 and TGRE_0.<br>0: Interrupt requests (TGIE) by TGEE bit disab<br>1: Interrupt requests (TGIE) by TGEE bit enabl |

|  |  |  |  |  |
|---|---|---|---|---|
| 2 | TGIE5U | 0 | R/W | TGR Interrupt Enable 5U<br>Enables or disables interrupt requests (TGIU_ CMFU5 bit when the CMFU5 bit in TSR_5 is s<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_ CMFV5 bit when the CMFV5 bit in TSR_5 is s<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 CMFW5 bit when the CMFW5 bit in TSR_5 is<br>0: Interrupt requests (TGIW_5) disabled<br>1: Interrupt requests (TGIW_5) enabled |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TCFD | 1 | R | Count Direction Flag<br>Status flag that shows the direction in which TC counts in channels 1 to 4.<br>In channel 0, bit 7 is reserved. It is always read 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 sho always be 1. |
| 5 | TCFU | 0 | R/(W)*1 | Underflow Flag<br>Status flag that indicates that TCNT underflow h occurred when channels 1 and 2 are set to pha counting mode. Only 0 can be written, for flag c<br>In channels 0, 3, and 4, bit 5 is reserved. It is al read as 0 and the write value should always be<br>[Setting condition]<br>• When the TCNT value underflows (changes H'0000 to H'FFFF)<br>[Clearing condition]<br>• When 0 is written to TCFU after reading TC |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | [Clearing condition]<br>• When 0 is written to TCFV after reading TC<br>In channel 4, when DTC is activated by TC<br>interrupt and the DISEL bit of MRB in DTC<br>flag is also cleared. |
| 3 | TGFD | 0 | R/(W)*1 | Input Capture/Output Compare Flag D<br>Status flag that indicates the occurrence of TG capture or compare match in channels 0, 3, ar 0 can be written, for flag clearing. In channels bit 3 is reserved. It is always read as 0 and the value should always be 0.<br>[Setting conditions]<br>• When TCNT = TGRD and TGRD is functio output compare register<br>• When TCNT value is transferred to TGRD capture signal and TGRD is functioning as capture register<br>[Clearing conditions]<br>• When DTC is activated by TGID interrupt a DISEL bit of MRB in DTC is 0<br>• When 0 is written to TGFD after reading TC |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | • When TCNT value is transferred to TGRC b capture signal and TGRC is functioning as i capture register<br>[Clearing conditions]<br>• When DTC is activated by TGIC interrupt ar DISEL bit of MRB in DTC is 0<br>• When 0 is written to TGFC after reading TG |
| 1 | TGFB | 0 | R/(W)*1 | Input Capture/Output Compare Flag B<br>Status flag that indicates the occurrence of TGF capture or compare match. Only 0 can be writte flag clearing.<br>[Setting conditions]<br>• When TCNT = TGRB and TGRB is function output compare register<br>• When TCNT value is transferred to TGRB b capture signal and TGRB is functioning as i capture register<br>[Clearing conditions]<br>• When DTC is activated by TGIB interrupt ar DISEL bit of MRB in DTC is 0<br>• When 0 is written to TGFB after reading TG |

capture register

[Clearing conditions]

- When DTC is activated by TGIA interrupt a
  DISEL bit of MRB in DTC is 0
- When 0 is written to TGFA after reading TC

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

2. If another flag setting condition occurs before writing 0 to the bit after reading flag will not be cleared by writing 0 to it once. In this case, read the bit as 1 ag write 0 to it.

| | | | | |
|---|---|---|---|---|
| | | | | These bits are always read as 1. The write valu<br>always be 1. |
| 5 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu<br>always be 0. |
| 1 | TGFF | 0 | R/(W)*[1] | Compare Match Flag F<br>Status flag that indicates the occurrence of com<br>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 TG |
| 0 | TGFE | 0 | R/(W)*[1] | Compare Match Flag E<br>Status flag that indicates the occurrence of com<br>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 TG |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed w

2. If another flag setting condition occurs before writing 0 to the bit after reading i flag will not be cleared by writing 0 to it once. In this case, read the bit as 1 aga write 0 to it.

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | These bits are always read as 0. The write valu<br>always be 0. |
| 2 | CMFU5 | 0 | R/(W)*1 | Compare Match/Input Capture Flag U5<br>Status flag that indicates the occurrence of TGR<br>input capture or compare match.<br>[Setting conditions]<br>• When TCNTU_5 = TGRU_5 and TGRU_5 i<br>functioning as output compare register<br>• When TCNTU_5 value is transferred to TGR<br>input capture signal and TGRU_5 is functio<br>input capture register<br>• When TCNTU_5 value is transferred to TGR<br>TGRU_5 is functioning as a register for mea<br>pulse width of the external input signal. The<br>timing is specified by the IOC bits in timer I/<br>register U_5 (TIORU_5)*2<br>[Clearing conditions]<br>• When DTC is activated by a TGIU_5 interru<br>DISEL bit of MRB in DTC is 0<br>• When 0 is written to CMFU5 after reading C |

input capture register

- When TCNTV_5 value is transferred to TGRV
TGRV_5 is functioning as a register for meas
pulse width of the external input signal. The t
timing is specified by the IOC bits in timer I/C
register V_5 (TIORV_5)*2

[Clearing conditions]

- When DTC is activated by a TGIV_5 interrup
DISEL bit of MRB in DTC is 0
- When 0 is written to CMFV5 after reading CM

input capture register

- When TCNTW_5 value is transferred to TG
  TGRW_5 is functioning as a register for me
  the pulse width of the external input signal.
  transfer timing is specified by the IOC bits i
  control register W_5 (TIORW_5)*2

[Clearing conditions]

- When DTC is activated by a TGIW_5 interru
  the DISEL bit of MRB in DTC is 0
- When 0 is written to CMFW5 after reading
  1

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

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).

| 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 valu always be 0. |
| 2 | TTSE | 0 | R/W | Timing Select E<br>Specifies the timing for transferring data from T to TGRE_0 when they are used together for bu operation.<br>In channels 3 and 4, bit 2 is reserved. It is alwa as 0 and the write value should always be 0. W using channel 0 in other than PWM mode, do n 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 T TGRB in each channel when they are used tog buffer operation. When using a channel in othe PWM mode, do not set this bit to1.<br>0: When compare match B occurs in each cha<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 T TGRA in each channel when they are used tog buffer operation. When using a channel in othe PWM mode, do not set this bit to 1.<br>0: When compare match A occurs in each cha<br>1: When TCNT is cleared in each channel |

|  |  |  |  |  |
|---|---|---|---|---|
| 7 to 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va<br>always be 0. |
| 3 | I2BE | 0 | R/W | Input Capture Enable<br>Specifies whether to include the TIOC2B pin i<br>TGRB_1 input capture conditions.<br>0: Does not include the TIOC2B pin in the TG<br>input capture conditions<br>1: Includes the TIOC2B pin in the TGRB_1 in<br>capture conditions |
| 2 | I2AE | 0 | R/W | Input Capture Enable<br>Specifies whether to include the TIOC2A pin i<br>TGRA_1 input capture conditions.<br>0: Does not include the TIOC2A pin in the TG<br>input capture conditions<br>1: Includes the TIOC2A pin in the TGRA_1 in<br>capture conditions |
| 1 | I1BE | 0 | R/W | Input Capture Enable<br>Specifies whether to include the TIOC1B pin i<br>TGRB_2 input capture conditions.<br>0: Does not include the TIOC1B pin in the TG<br>input capture conditions<br>1: Includes the TIOC1B pin in the TGRB_2 in<br>capture conditions |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CE0A | 0 | R/W | Clear Enable 0A<br>Enables or disables counter clearing when th<br>flag of TSR_0 in the MTU2 is set.<br>0: Disables counter clearing by the TGFA flag<br>1: Enables counter clearing by the TGFA flag |
| 6 | CE0B | 0 | R/W | Clear Enable 0B<br>Enables or disables counter clearing when th<br>flag of TSR_0 in the MTU2 is set.<br>0: Disables counter clearing by the TGFB flag<br>1: Enables counter clearing by the TGFB flag |
| 5 | CE0C | 0 | R/W | Clear Enable 0C<br>Enables or disables counter clearing when th<br>flag of TSR_0 in the MTU2 is set.<br>0: Disables counter clearing by the TGFC flag<br>1: Enables counter clearing by the TGFC flag |
| 4 | CE0D | 0 | R/W | Clear Enable 0D<br>Enables or disables counter clearing when th<br>flag of TSR_0 in the MTU2 is set.<br>0: Disables counter clearing by the TGFD flag<br>1: Enables counter clearing by the TGFD flag |
| 3 | CE1A | 0 | R/W | Clear Enable 1A<br>Enables or disables counter clearing when th<br>flag of TSR_1 in the MTU2 is set.<br>0: Disables counter clearing by the TGFA flag<br>1: Enables counter clearing by the TGFA flag |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | 0: Disables counter clearing by the TGFA flag<br>1: Enables counter clearing by the TGFA flag i |
| 0 | CE2B | 0 | R/W | Clear Enable 2B<br>Enables or disables counter clearing when the flag of TSR_2 in the MTU2 is set.<br>0: Disables counter clearing by the TGFB flag<br>1: Enables counter clearing by the TGFB flag i |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15, 14 | BF[1:0] | 00 | R/W | TADCOBRA_4/TADCOBRB_4 Transfer Timi<br>Select the timing for transferring data from TADCOBRA_4 and TADCOBRB_4 to TADC 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 va always be 0. |
| 7 | UT4AE | 0 | R/W | Up-Count TRG4AN Enable<br>Enables or disables A/D converter start reque (TRG4AN) during TCNT_4 up-count operatio<br>0: A/D converter start requests (TRG4AN) dis during TCNT_4 up-count operation<br>1: A/D converter start requests (TRG4AN) en during TCNT_4 up-count operation |
| 6 | DT4AE | 0* | R/W | Down-Count TRG4AN Enable<br>Enables or disables A/D converter start reque (TRG4AN) during TCNT_4 down-count opera<br>0: A/D converter start requests (TRG4AN) dis during TCNT_4 down-count operation<br>1: A/D converter start requests (TRG4AN) en during TCNT_4 down-count operation |

| | | | | |
|---|---|---|---|---|
| | | | | (TRG4BN) during TCNT_4 down-count operati<br>0: A/D converter start requests (TRG4BN) disa during TCNT_4 down-count operation<br>1: A/D converter start requests (TRG4BN) ena 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 reque (TRG4AN) with TGIA_3 interrupt skipping oper<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 reque (TRG4AN) with TCIV_4 interrupt skipping oper<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 reque (TRG4BN) with TGIA_3 interrupt skipping oper<br>0: Does not link with TGIA_3 interrupt skipping<br>1: Links with TGIA_3 interrupt skipping |

interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE
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/
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 t<br>set register. |
| 0 | 1 | Transfers data from the cycle set buffer register to the cycle<br>register at the crest of the TCNT_4 count.*[1] |
| 1 | 0 | Transfers data from the cycle set buffer register to the cycle<br>register at the trough of the TCNT_4 count.*[2] |
| 1 | 1 | Transfers data from the cycle set buffer register to the cycle<br>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 w
crest of the TCNT_4 count is reached in complementary PWM mode, when c
match occurs between TCNT_3 and TGRA_3 in reset-synchronized PWM mo
when compare match occurs between TCNT_4 and TGRA_4 in PWM mode
normal operation mode.

2. These settings are prohibited when complementary PWM mode is not selecte

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | F |

Note: TADCORA_4 and TADCORB_4 must not be accessed in eight bits; they should always be accessed in 16 b

### 10.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOB and TADCOBRB_4)

TADCOBRA_4 and TADCOBRB_4 are 16-bit readable/writable registers. When the cre trough of the TCNT_4 count is reached, these register values are transferred to TADCOR 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 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | |
| Initial value: | 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 | F |

Note: TADCOBRA_4 and TADCOBRB_4 must not be accessed in eight bits; they should always be accessed in 1

### 10.3.14 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers. The MTU2 has 21 TGR registe channel 0, two each for channels 1 and 2, four each for channels 3 and 4, and three for c

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture re TGRC and TGRD for channels 0, 3, and 4 can also be designated for operation as buffer 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 TGRE_0 value, an A/D converter start request can be issued. TGRF can also be designa 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 externa width measurement registers.

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

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.

| 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 TC<br>If 0 is written to the CST bit during operation w TIOC pin designated for output, the counter st the TIOC pin output compare output level is re TIOR is written to when the CST bit is cleared pin output level will be changed to the set initia value.<br>0: TCNT_4 and TCNT_3 count operation is sto<br>1: TCNT_4 and TCNT_3 performs count opera |
| 6 | CST3 | 0 | R/W | |
| 5 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu always be 0. |

1: TCNT_2 to TCNT_0 performs count operat

- 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 va 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 |

| Bit | Bit Name | 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 operatio independent of or synchronized with other cha<br>When synchronous operation is selected, the T synchronous presetting of multiple channels, a synchronous clearing by counter clearing on ar channel, are possible.<br>To set synchronous operation, the SYNC bits f least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC TCNT clearing source must also be set by mea bits CCLR0 to CCLR2 in TCR.<br>0: TCNT_4 and TCNT_3 operate independentl presetting/clearing is unrelated to other char<br>1: TCNT_4 and TCNT_3 performs synchronou operation TCNT synchronous presetting/synchronous is possible |
| 5 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu always be 0. |

synchronous clearing, in addition to the SYNC
TCNT clearing source must also be set by me
bits CCLR0 to CCLR2 in TCR.

0: TCNT_2 to TCNT_0 operates independent
presetting /clearing is unrelated to other ch

1: TCNT_2 to TCNT_0 performs synchronous
TCNT synchronous presetting/synchronous
is possible

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | SCH0 | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_0 in the M<br>0: Does not specify synchronous start for TCN the MTU2<br>1: Specifies synchronous start for TCNT_0 in t<br>[Clearing condition]<br>• When 1 is set to the CST0 bit of TSTR in M while SCH0 = 1 |
| 6 | SCH1 | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_1 in the M<br>0: Does not specify synchronous start for TCN the MTU2<br>1: Specifies synchronous start for TCNT_1 in t<br>[Clearing condition]<br>• When 1 is set to the CST1 bit of TSTR in M while SCH1 = 1 |
| 5 | SCH2 | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_2 in the M<br>0: Does not specify synchronous start for TCN the MTU2<br>1: Specifies synchronous start for TCNT_2 in t<br>[Clearing condition]<br>• When 1 is set to the CST2 bit of TSTR in M while SCH2 = 1 |

| | | | | |
|---|---|---|---|---|
| 3 | SCH4 | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_4 in the<br>0: Does not specify synchronous start for TCN the MTU2<br>1: Specifies synchronous start for TCNT_4 in<br>[Clearing condition]<br>• When 1 is set to the CST4 bit of TSTR in while SCH4 = 1 |
| 2 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s always be 0. |
| 1 | SCH3S | 0 | R/(W)* | Synchronous Start<br>Controls synchronous start of TCNT_3S in th<br>0: Does not specify synchronous start for TCN the MTU2S<br>1: Specifies synchronous start for TCNT_3S i MTU2S<br>[Clearing condition]<br>• When 1 is set to the CST3 bit of TSTRS i while SCH3S = 1 |

Note: * Only 1 can be written to set the register.

| 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 va<br>always be 0. |
| 0 | RWE | 1 | R/W | Read/Write Enable<br>Enables or disables access to the registers w<br>write-protection capability against accidental<br>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 rea<br>RWE = 1 |

- Registers and counters having write-protection capability against accidental modifica
  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, TOC
  TOCR2, TGCR, TCDR, TDDR, TCNT_3, and TCNT4.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write valu always be 1. |
| 5 | OE4D | 0 | R/W | Master Enable TIOC4D<br>This bit enables/disables the TIOC4D pin MTU<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 MTU<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 MTU<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 MTU<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 MTU<br>0: MTU2 output is disabled (inactive level)*<br>1: MTU2 output is enabled |

### 10.3.20 Timer Output Control Register 1 (TOCR1)

TOCR1 is an 8-bit readable/writable register that enables/disables PWM synchronized t
output in complementary PWM mode/reset synchronized PWM mode, and controls out
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 t

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s always be 0. |
| 6 | PSYE | 0 | R/W | PWM Synchronous Output Enable<br>This bit selects the enable/disable of toggle o 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 va always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | be used for the output level in complementary 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 synchronized PWM mode/complementary PW<br>See table 10.30. |
| 0 | OLSP | 0 | R/W | Output Level Select P*[2]<br>This bit selects the positive phase output level synchronized PWM mode/complementary PW<br>See table 10.31. |

Notes: 1. Setting the TOCL bit to 1 prevents accidental modification when the CPU goes 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 elaps dead time after count start.

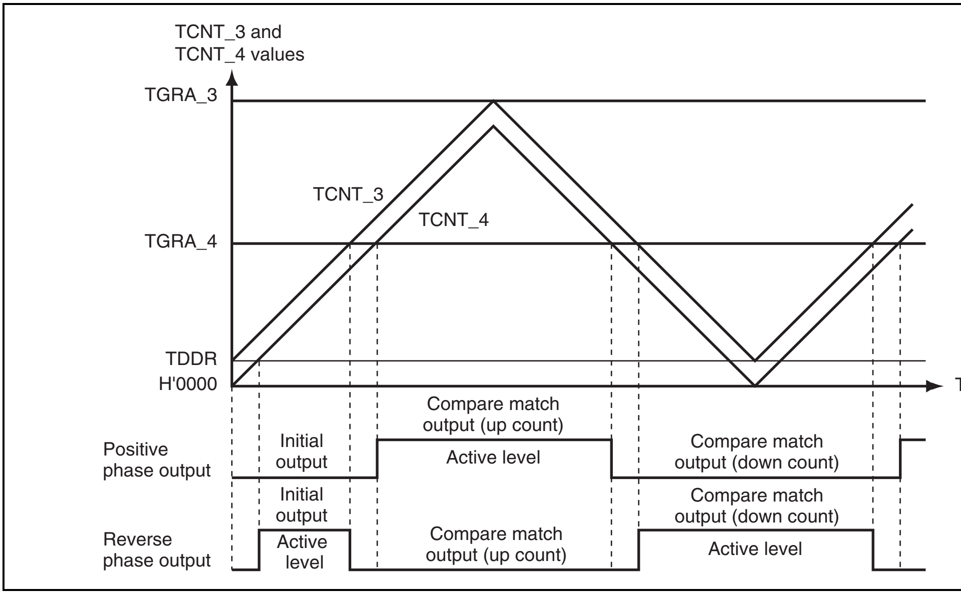


**Figure 10.2 Complementary PWM Mode Output Level Example**

| | | | | |
|---|---|---|---|---|
| 7, 6 | BF[1:0] | 00 | R/W | TOLBR Buffer Transfer Timing Select<br>These bits select the timing for transferring dat<br>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 r<br>synchronized PWM mode/complementary PW<br>See table 10.33. |
| 4 | OLS3P | 0 | R/W | Output Level Select 3P*<br>This bit selects the output level on TIOC4B in r<br>synchronized PWM mode/complementary PW<br>See table 10.34. |
| 3 | OLS2N | 0 | R/W | Output Level Select 2N*<br>This bit selects the output level on TIOC4C in r<br>synchronized PWM mode/complementary PW<br>See table 10.35. |
| 2 | OLS2P | 0 | R/W | Output Level Select 2P*<br>This bit selects the output level on TIOC4A in r<br>synchronized PWM mode/complementary PW<br>See table 10.36. |
| 1 | OLS1N | 0 | R/W | Output Level Select 1N*<br>This bit selects the output level on TIOC3D in r<br>synchronized PWM mode/complementary PW<br>See table 10.37. |
| 0 | OLS1P | 0 | R/W | Output Level Select 1P*<br>This bit selects the output level on TIOC3B in r<br>synchronized PWM mode/complementary PW<br>See table 10.38. |

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

| | | | |
|---|---|---|---|
| | | trough of the TCNT_4 count. | |
| 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 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 10.36 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 e 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 |

|  |  |  |  |  |
|---|---|---|---|---|
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va always be 0. |
| 5 | OLS3N | 0 | R/W | Specifies the buffer value to be transferred to OLS3N bit in TOCR2. |
| 4 | OLS3P | 0 | R/W | Specifies the buffer value to be transferred to OLS3P bit in TOCR2. |
| 3 | OLS2N | 0 | R/W | Specifies the buffer value to be transferred to OLS2N bit in TOCR2. |
| 2 | OLS2P | 0 | R/W | Specifies the buffer value to be transferred to OLS2P bit in TOCR2. |
| 1 | OLS1N | 0 | R/W | Specifies the buffer value to be transferred to OLS1N bit in TOCR2. |
| 0 | OLS1P | 0 | R/W | Specifies the buffer value to be transferred to OLS1P bit 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 f brushless DC motor control in reset-synchronized PWM mode/complementary PWM mo 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 sh always be 1. |
| 6 | BDC | 0 | R/W | Brushless DC Motor<br>This bit selects whether to make the functions register (TGCR) effective or ineffective.<br>0: Ordinary output<br>1: Functions of this register are made effective |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | This bit selects whether the level output or the<br>synchronized PWM/complementary PWM out<br>the positive pin (TIOC3B, TIOC4A, and TIOC<br>output.<br>0: Level output<br>1: Reset synchronized PWM/complementary<br>output |
| 3 | FB* | 0 | R/W | External Feedback Signal Enable<br>This bit selects whether the switching of the o<br>the positive/reverse phase is carried out autor<br>with the MTU2/channel 0 TGRA, TGRB, TGR<br>capture signals or by writing 0 or 1 to bits 2 to<br>TGCR.<br>0: Output switching is external input (Input so<br>channel 0 TGRA, TGRB, TGRC input captu<br>1: Output switching is carried out by software<br>UF, VF, WF settings). |
| 2 | WF | 0 | R/W | Output Phase Switch 2 to 0<br>These bits set the positive phase/negative ph<br>phase on or off state. The setting of these bits<br>only when the FB bit in this register is set to 1<br>case, the setting of bits 2 to 0 is a substitute f<br>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.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | ON | OFF | OFF | OFF | ON | C |
| 1 | 0 | OFF | OFF | ON | ON | OFF | C |
| | 1 | OFF | OFF | OFF | OFF | OFF | C |

### 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Note: Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 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 |

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 c
value as the TCDR register value. This register is constantly compared with the TCNTS
complementary PWM mode, and when a match occurs, the TCNTS counter switches di
(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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | |
| Initial value: | 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 |

Note: Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

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 an 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 skippir 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 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. |

| | | | | |
|---|---|---|---|---|
| 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 c these bits is incremented every time a TGIA_3 occurs.<br>[Clearing conditions]<br>• When the 3ACNT2 to 3ACNT0 value in TIT matches the 3ACOR2 to 3ACOR0 value in<br>• When the T3AEN bit in TITCR is cleared to<br>• When the 3ACOR2 to 3ACOR0 bits in TITC 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 c these bits is incremented every time a TCIV_4 occurs.<br>[Clearing conditions]<br>• When the 4VCNT2 to 4VCNT0 value in TIT matches the 4VCOR2 to 4VCOR2 value in<br>• When the T4VEN bit in TITCR is cleared to<br>• When the 4VCOR2 to 4VCOR2 bits in TITC cleared to 0 |

Note: To clear the TITCNT, clear the T3AEN and T4VEN bits in TITCR to 0.

| Bit | Bit Name | Value | R/W | Description |
|---|---|---|---|---|
| 7 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va always be 0. |
| 1, 0 | BTE[1:0] | 00 | R/W | These bits enable or disable transfer from the registers* used in complementary PWM mod temporary registers and specify whether to lir 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 and does not link the transfer with interrupt skipping operat |
| 0 | 1 | Disables transfer from the buffer registers to the temporary |
| 1 | 0 | Links transfer from the buffer registers to the temporary reg interrupt skipping operation.*2 |
| 1 | 1 | Setting prohibited |

Notes: 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For det to section 10.4.8, Complementary PWM Mode.

2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared t timer interrupt skipping set register (TITCR) or the skipping count set bits (3A 4VCOR) in TITCR are cleared to 0)), be sure to disable link of buffer transfer interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (T 0). If link with interrupt skipping is enabled while interrupt skipping is disabled transfer will not be performed.

| Bit | Bit Name | Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu<br>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 TD |

Note: * TDDR must be set to 1 or a larger value.

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_ compare match in complementary PWM mod<br>0: Does not clear counters at TGRA_3 compa<br>1: Clears counters at TGRA_3 compare matc<br>[Setting condition]<br>• When 1 is written to CCE after reading C( |
| 6 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va always be 0. |

Counter clearing synchronized with the MTU2
disabled by the SCC bit setting only when sync
clearing occurs outside the Tb interval at the tr
When synchronous clearing occurs in the Tb in
the trough including the period immediately afte
TCNT_3 and TCNT_4 start operation, TCNT_3
TCNT_4 in the MTU2S are cleared.

For the Tb interval at the trough in complemen
PWM mode, see figure 10.40.

In the MTU2, this bit is reserved. It is always re
and the write value should always be 0.

0: Enables clearing of TCNT_3 and TCNT_4 in
MTU2S by MTU2–MTU2S synchronous clea
operation

1: Disables clearing of TCNT_3 and TCNT_4 in
MTU2S by MTU2–MTU2S synchronous clea
operation

[Setting condition]

- When 1 is written to SCC after reading SCC

trough immediately after TCNT_3 and TCNT_
operation.

For the Tb interval at the trough in compleme
PWM mode, see figure 10.40.

0: Outputs the initial value specified in TOCR

1: Suppresses initial output

[Setting condition]

- When 1 is written to WRE after reading W

Note: * Do not set to 1 when complementary PWM mode is not selected.

### 10.3.33 Bus Master Interface

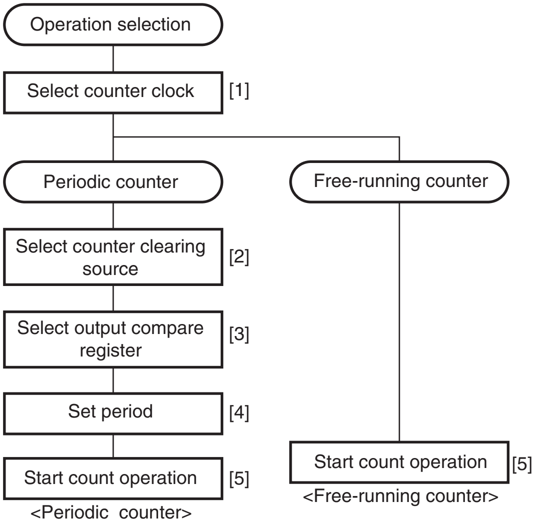
The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer
buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register
timer A/D converter start request control register (TADCR), timer A/D converter start re
cycle set registers (TADCOR), and timer A/D converter start request cycle set buffer reg
(TADCOBR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read
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
16-bit data bus, so 16-bit read/writes and 8-bit read/writes are both possible.

When one of bits CST0 to CST4 in TSTR or bits CSTU5, CSTV5, and CSTW5 in TSTR
to 1, the TCNT counter for the corresponding channel begins counting. TCNT can operat
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.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] For periodic counter operation, select the TGR to be used as the TCNT clearing source with bits CCLR2 to CCLR0 in TCR.

[3] Designate the TGR selected in [2] as an output compare register by means of TIOR.

[4] Set the periodic counter cycle in the TGR selected in [2].

[5] Set the CST bit in TSTR to 1 to start the counter operation.

**Figure 10.4 Example of Counter Operation Setting Procedure**

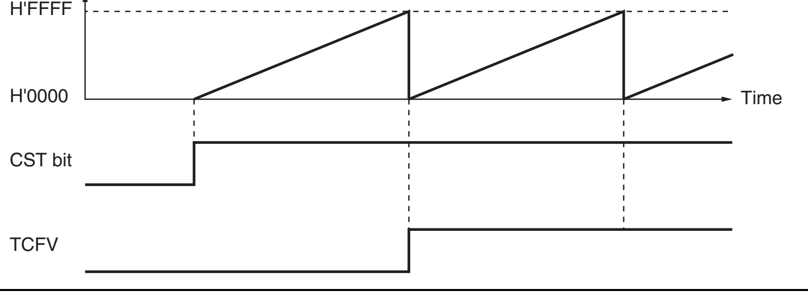


**Figure 10.5 Free-Running Counter Operation**

When compare match is selected as the TCNT clearing source, the TCNT counter fo relevant channel performs periodic count operation. The TGR register for setting the designated as an output compare register, and counter clearing by compare match is by means of bits CCLR0 to CCLR2 in TCR. After the settings have been made, TCN up-count operation as a periodic counter when the corresponding bit in TSTR is set t the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the MTU2 requ interrupt. After a compare match, TCNT starts counting up again from H'0000.

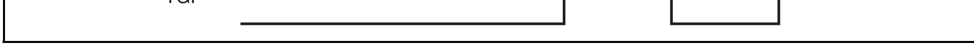


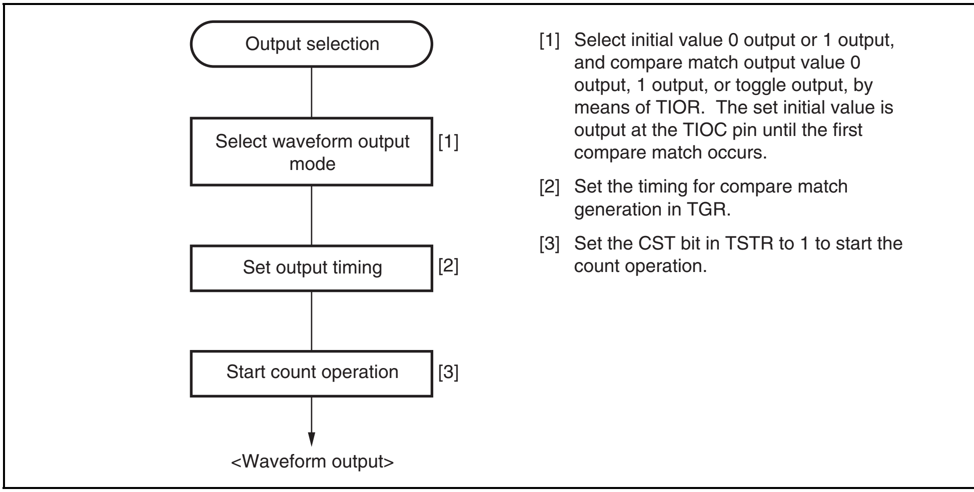
Figure 10.6 Periodic Counter Operation

**Waveform Output by Compare Match:**

The MTU2 can perform 0, 1, or toggle output from the corresponding output pin using co 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 compa



[1] Select initial value 0 output or 1 output, and compare match output value 0 output, 1 output, or toggle output, by means of TIOR. The set initial value is output at the TIOC pin until the first compare match occurs.

[2] Set the timing for compare match generation in TGR.

[3] Set the CST bit in TSTR to 1 to start the count operation.

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

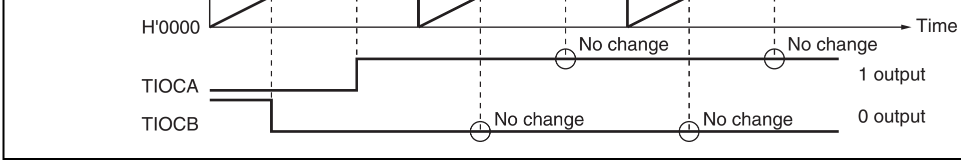


**Figure 10.8   Example of 0 Output/1 Output Operation**

Figure 10.9 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clear compare match B), and settings have been made such that the output is toggled by bo compare match A and compare match B.

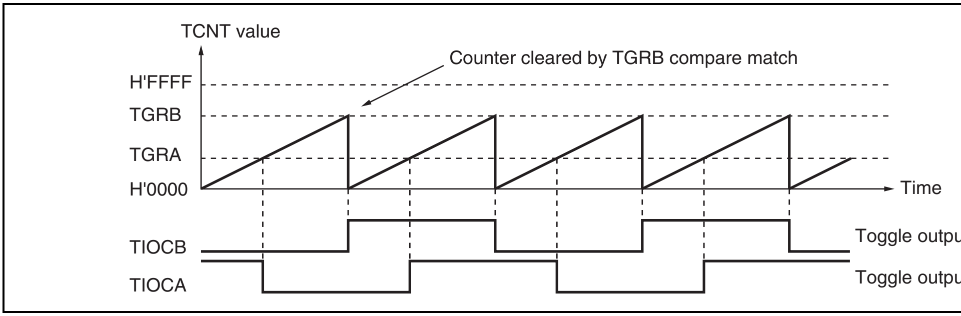
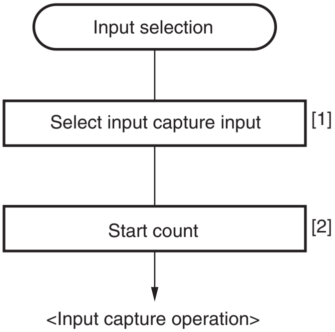


**Figure 10.9   Example of Toggle Output Operation**

1. Example of Input Capture Operation Setting Procedure

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



[1] Designate TGR as an input capture register by means of TIOR, and select rising edge, falling edge, or both edges as the input capture source and input signal edge.

[2] Set the CST bit in TSTR to 1 to start the count operation.

**Figure 10.10 Example of Input Capture Operation Setting Procedure**

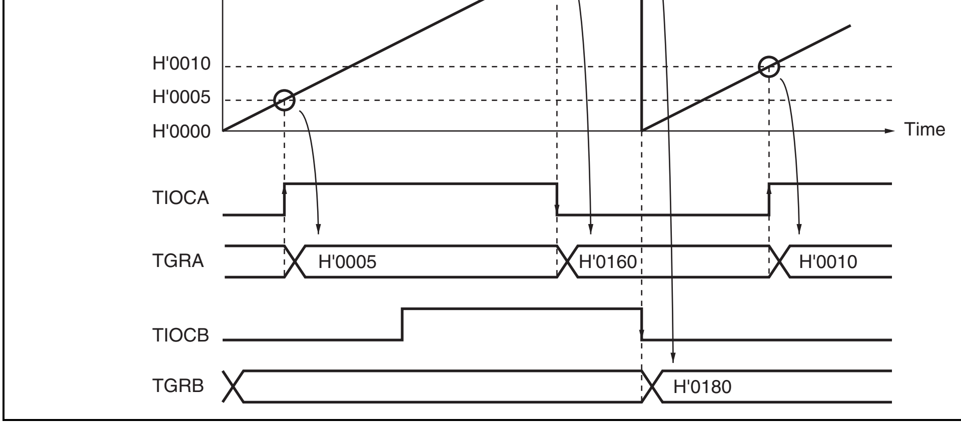
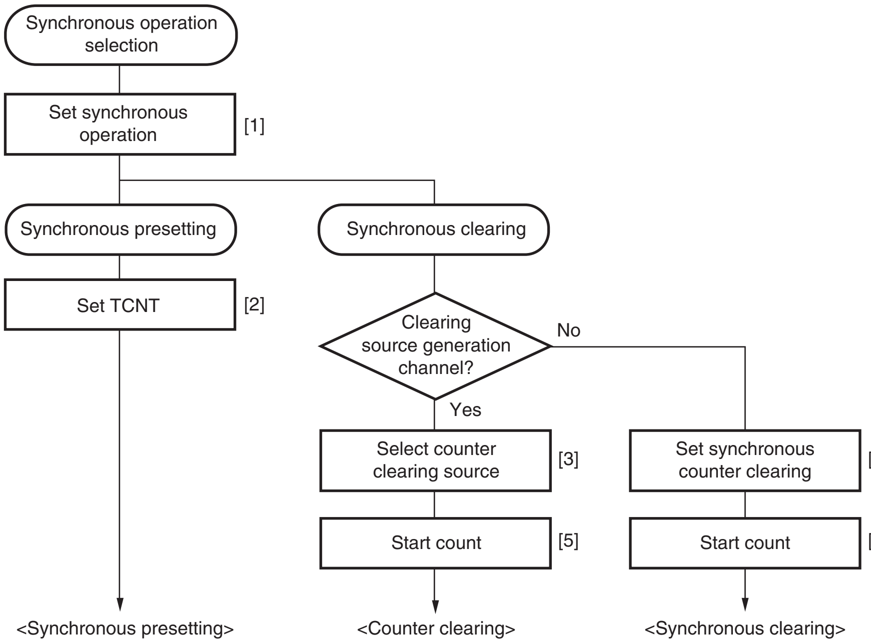


**Figure 10.11 Example of Input Capture Operation**

**Example of Synchronous Operation Setting Procedure**

Figure 10.12 shows an example of the synchronous operation setting procedure.


Synchronous operation selection
Set synchronous operation
[1]
Synchronous presetting
Synchronous clearing
Set TCNT
[2]
Clearing source generation channel?
No
Yes
Select counter clearing source
[3]
Set synchronous counter clearing
Start count
[5]
Start count
<Synchronous presetting>
<Counter clearing>
<Synchronous clearing>

[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**

For details of PWM modes, see section 10.4.5, PWM Modes.

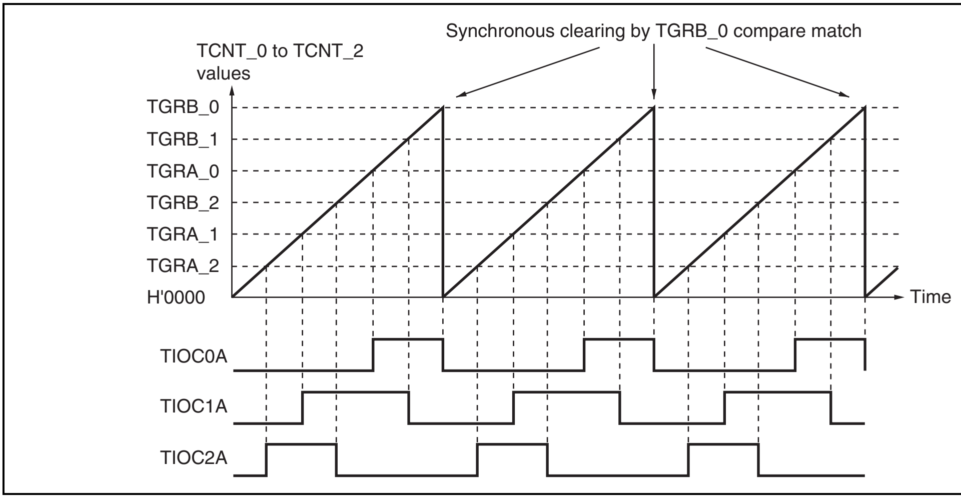


**Figure 10.13 Example of Synchronous Operation**

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 c transferred to the timer general register.

  This operation is illustrated in figure 10.14.

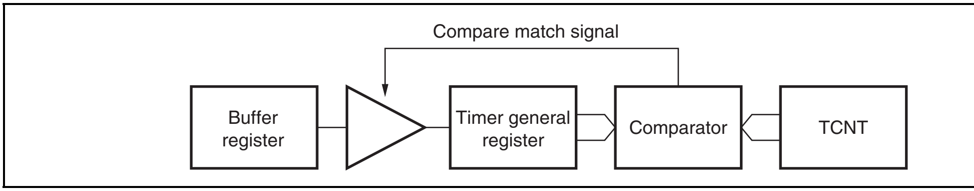
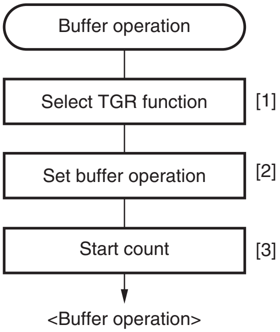


**Figure 10.14 Compare Match Buffer Operation**

**Figure 10.15 Input Capture Buffer Operation**

**Example of Buffer Operation Setting Procedure:** Figure 10.16 shows an example of t
operation setting procedure.



[1] Designate TGR as an input capture register or output compare register by means of TIOR.

[2] Designate TGR for buffer operation with bits BFA and BFB in TMDR.

[3] Set the CST bit in TSTR to 1 start the count operation.

**Figure 10.16 Example of Buffer Operation Setting Procedure**

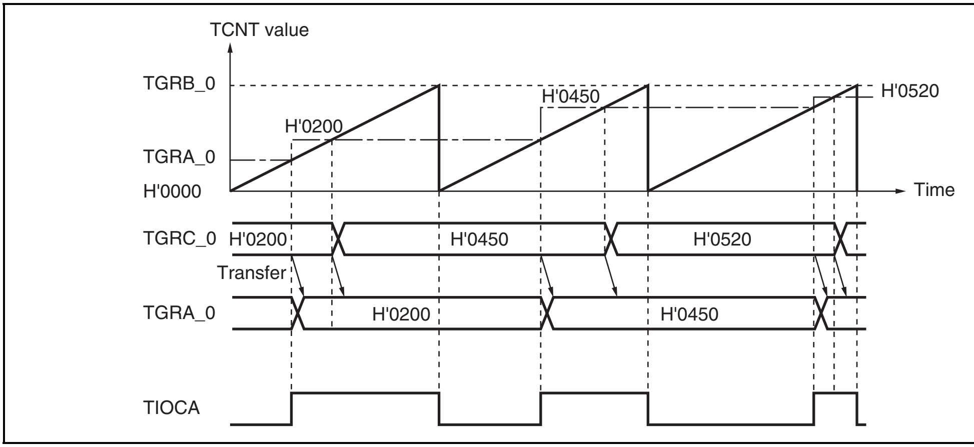


**Figure 10.17   Example of Buffer Operation (1)**

2. When TGR is an input capture register

   Figure 10.18 shows an operation example in which TGRA has been designated as an 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 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.

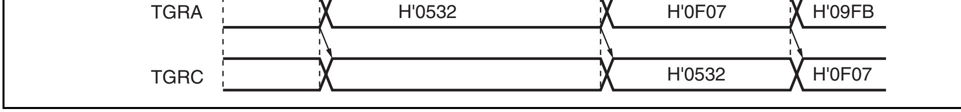


**Figure 10.18 Example of Buffer Operation (2)**

**Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Operation:** The timing for transfer from buffer registers to timer general registers can b in PWM mode 1 or 2 for channel 0 or in PWM mode 1 for channels 3 and 4 by setting th operation transfer mode registers (TBTM_0, TBTM_3, and TBTM_4). Either compare r (initial setting) or TCNT clearing can be selected for the transfer timing. TCNT clearing 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 CC 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 chan buffer operation is designated for TGRA_0 and TGRC_0. The settings used in this exam TCNT_0 clearing by compare match B, 1 output at compare match A, and 0 output at co match B. The TTSA bit in TBTM_0 is set to 1.

**Figure 10.19 Example of Buffer Operation When TCNT_0 Clearing is Selecte 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 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 inval counters operates independently in phase counting mode.

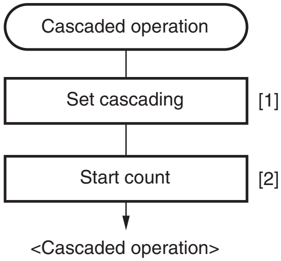
**Table 10.44 Cascaded Combinations**

| 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, addi input capture input pins can be specified by the input capture control register (TICCR). F capture in cascade connection, refer to section 10.7.22, Simultaneous Capture of TCNT_ TCNT_2 in Cascade Connection.

| | | |
|---|---|---|
| _ | 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 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_ overflow/underflow counting has been set for TCNT_1 and phase counting mode has be designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow

TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in T have selected the TIOC1A rising edge for the input capture timing while the IOA0 to IO 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 TG input capture condition. For the TGRA_2 input capture condition, the TIOC2A rising edg

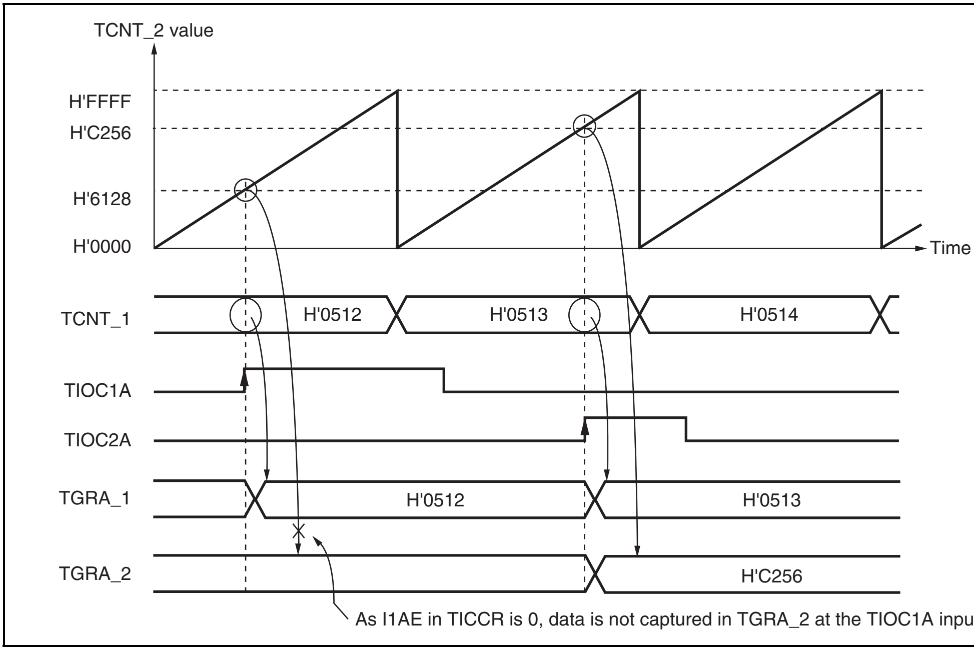


**Figure 10.22 Cascaded Operation Example (b)**

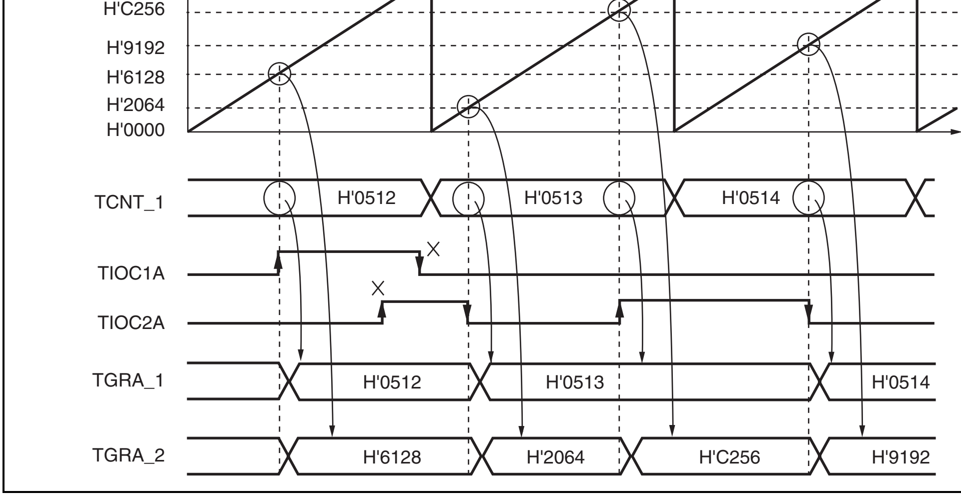


**Figure 10.23 Cascaded Operation Example (c)**

**Cascaded Operation Example (d):** Figure 10.24 illustrates the operation when TCNT_
TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the
pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in
have selected TGRA_0 compare match or input capture occurrence for the input capture
while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the i
capture timing.

Under these conditions, as TIOR_1 has selected TGRA_0 compare match or input captu
occurrence for the input capture timing, the TIOC2A edge is not used for TGRA_1 inpu
condition although the I2AE bit in TICCR has been set to 1.

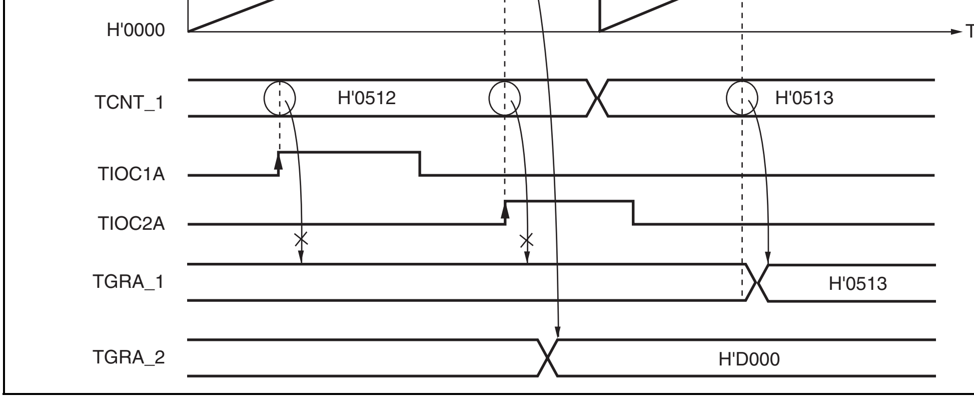


**Figure 10.24 Cascaded Operation Example (d)**

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 T TGRC with TGRD. The output specified by bits IOA0 to IOA3 and IOC0 to IOC3 in output from the TIOCA and TIOCC pins at compare matches A and C, and the outpu specified by bits IOB0 to IOB3 and IOD0 to IOD3 in TIOR is output at compare ma and D. The initial output value is the value set in TGRA or TGRC. If the set values o 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

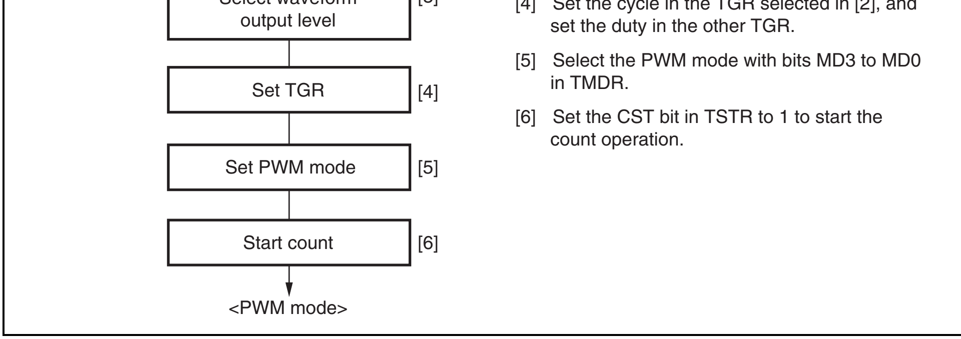
   PWM output is generated using one TGR as the cycle register and the others as duty The output specified in TIOR is performed by means of compare matches. Upon cou clearing by a synchronization register compare match, the output value of each pin is value set in TIOR. If the set values of the cycle and duty registers are identical, the o value does not change when a compare match occurs.

   In PWM mode 2, a maximum 8-phase PWM output is possible in combination use w synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10.46.

|  |  |  |  |
|---|---|---|---|
| 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 per



[4] Set the cycle in the TGR selected in [2], and set the duty in the other TGR.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

[6] Set the CST bit in TSTR to 1 to start the count operation.

**Figure 10.25 Example of PWM Mode Setting Procedure**

**Examples of PWM Mode Operation:** Figure 10.26 shows an example of PWM mode operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for th 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 are used as the duty levels.

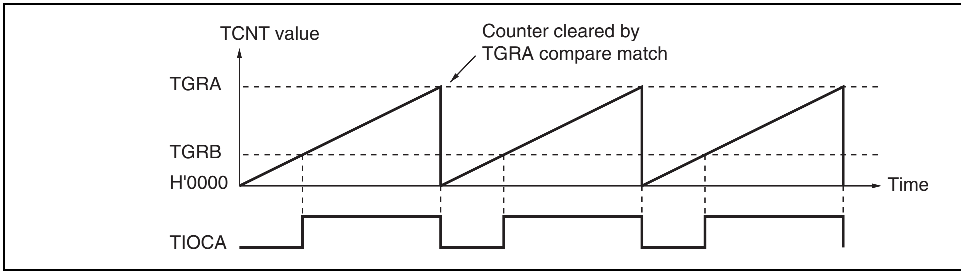


**Figure 10.26 Example of PWM Mode Operation (1)**

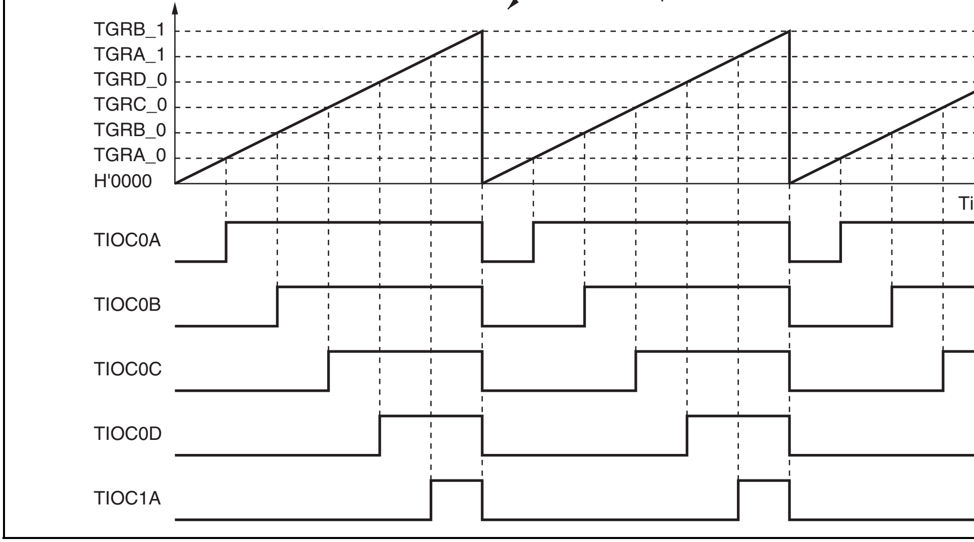


**Figure 10.27 Example of PWM Mode Operation (2)**

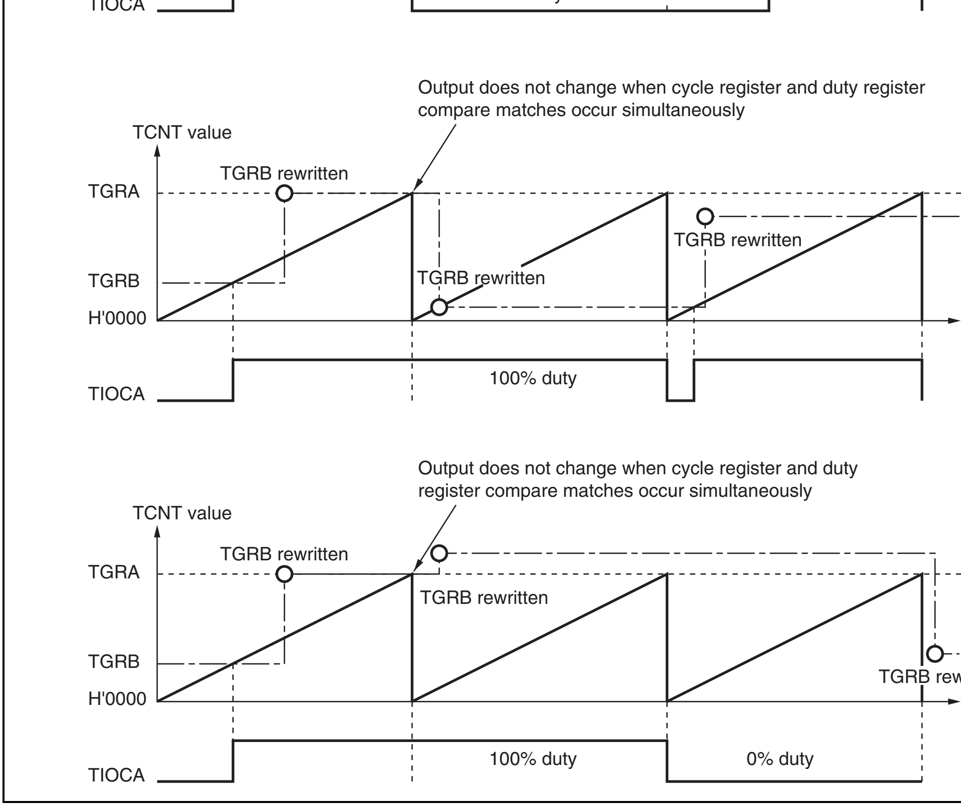

TIOCA
Output does not change when cycle register and duty register compare matches occur simultaneously
TCNT value
TGRB rewritten
TGRA
TGRB rewritten
TGRB
TGRB rewritten
H'0000
100% duty
TIOCA
Output does not change when cycle register and duty register compare matches occur simultaneously
TCNT value
TGRB rewritten
TGRA
TGRB rewritten
TGRB
TGRB rew
H'0000
100% duty
0% duty
TIOCA

**Figure 10.28 Example of PWM Mode Operation (3)**

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 when TCNT is counting down, the TCFU flag is set.

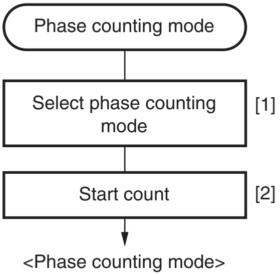
The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether counting up or down.

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 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**

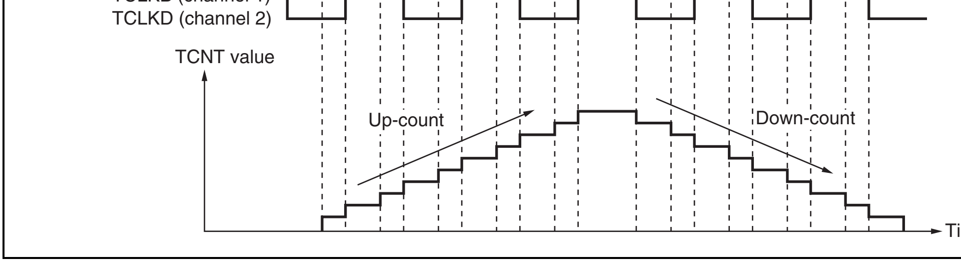


**Figure 10.30 Example of Phase Counting Mode 1 Operation**

**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

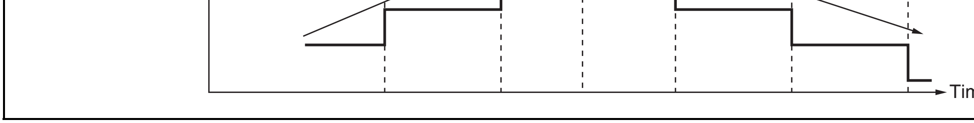


Figure 10.31 Example of Phase Counting Mode 2 Operation

**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

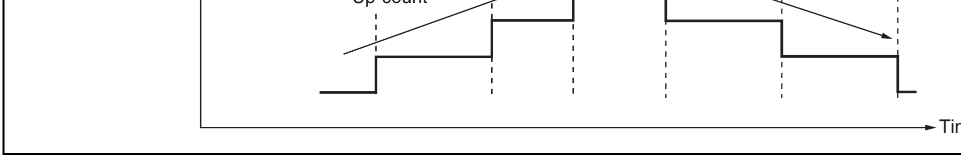


Figure 10.32 Example of Phase Counting Mode 3 Operation

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

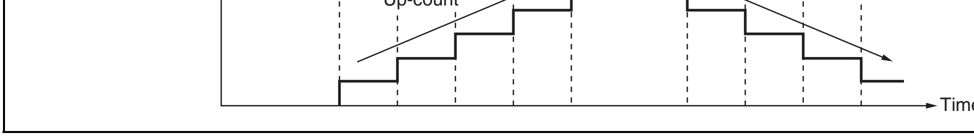


**Figure 10.33 Example of Phase Counting Mode 4 Operation**

**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

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 TG
TGRC_0 compare matches are selected as the input capture source and store the up/dow
values for the control periods.

This procedure enables the accurate detection of position and speed.

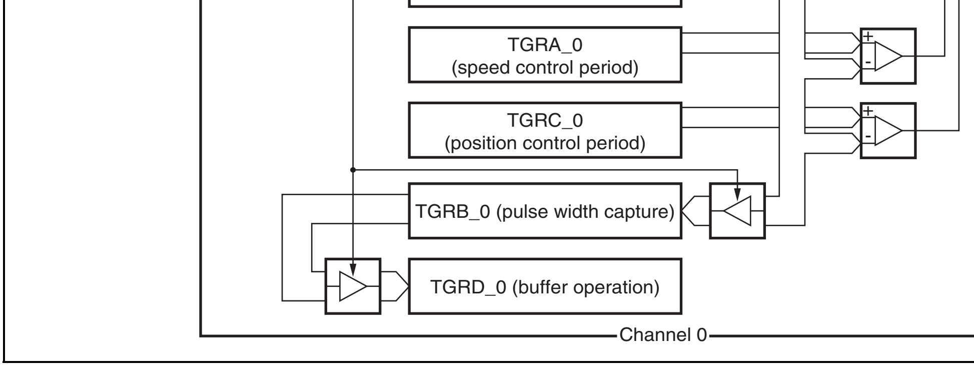

TGRA_0
(speed control period)
TGRC_0
(position control period)
TGRB_0 (pulse width capture)
TGRD_0 (buffer operation)
+
-
+
-
Channel 0

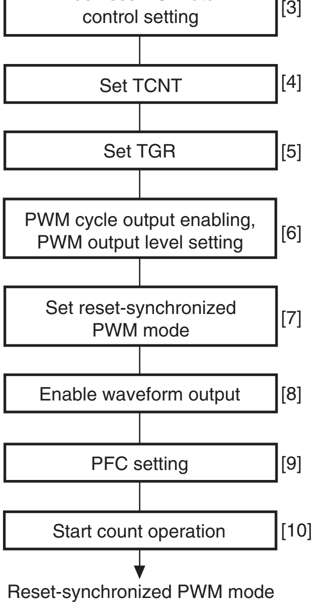
**Figure 10.34 Phase Counting Mode Application Example**

### 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 ou |
| 4 | TIOC4A | PWM output pin 2 |
|  | TIOC4C | PWM output pin 2' (negative-phase waveform of PWM ou |
|  | TIOC4B | PWM output pin 3 |
|  | TIOC4D | PWM output pin 3' (negative-phase waveform of PWM ou |

### 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 TIOC3 |
| TGRA_4 | Sets the turning point for PWM waveform output by the TIOC4A and TIOC4 |
| TGRB_4 | Sets the turning point for PWM waveform output by the TIOC4B and TIOC4 |


control setting
[3]
Set TCNT
[4]
Set TGR
[5]
PWM cycle output enabling, PWM output level setting
[6]
Set reset-synchronized PWM mode
[7]
Enable waveform output
[8]
PFC setting
[9]
Start count operation
[10]
Reset-synchronized PWM mode

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

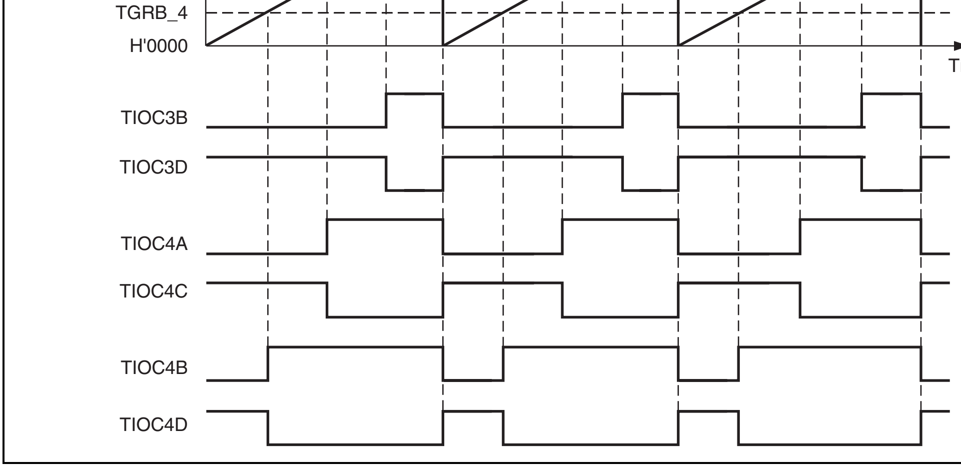


**Figure 10.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)**

used.

A function to directly cut off the PWM output by using an external signal is supported as 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 p |
| | TIOC3B | PWM output pin 1 |
| | TIOC3C | I/O port* |
| | TIOC3D | PWM output pin 1'<br>(non-overlapping negative-phase waveform of PWM o<br>PWM output without non-overlapping interval is also a |
| 4 | TIOC4A | PWM output pin 2 |
| | TIOC4B | PWM output pin 3 |
| | TIOC4C | PWM output pin 2'<br>(non-overlapping negative-phase waveform of PWM o<br>PWM output without non-overlapping interval is also a |
| | TIOC4D | PWM output pin 3'<br>(non-overlapping negative-phase waveform of PWM o<br>PWM output without non-overlapping interval is also a |

Note: * Avoid setting the TIOC3C pin as a timer I/O pin in the complementary PWM m

| | | | |
|---|---|---|---|
| | | register | |
| 4 | TCNT_4 | Up-count start, initialized to H'0000 | Maskable by TRW setting* |
| | TGRA_4 | PWM output 2 compare register | Maskable by TRW setting* |
| | TGRB_4 | PWM output 3 compare register | Maskable by TRW setting* |
| | TGRC_4 | PWM output 2/TGRA_4 buffer register | Always readable/ |
| | TGRD_4 | PWM output 3/TGRB_4 buffer register | Always readable/ |
| Timer dead time data register (TDDR) | | Set TCNT_4 and TCNT_3 offset value (dead time value) | Maskable by TRW setting* |
| Timer cycle data register (TCDR) | | Set TCNT_4 upper limit value (1/2 carrier cycle) | Maskable by TRW setting* |
| Timer cycle buffer register (TCBR) | | TCDR buffer register | Always readable/ |
| Subcounter (TCNTS) | | Subcounter for dead time generation | Read-only |
| Temporary register 1 (TEMP1) | | PWM output 1/TGRB_3 temporary register | Not readable/writ |
| Temporary register 2 (TEMP2) | | PWM output 2/TGRA_4 temporary register | Not readable/writ |
| Temporary register 3 (TEMP3) | | PWM output 3/TGRB_4 temporary register | Not readable/writ |

Note: * Access can be enabled or disabled according to the setting of bit 0 (RWE) in (timer read/write enable register).

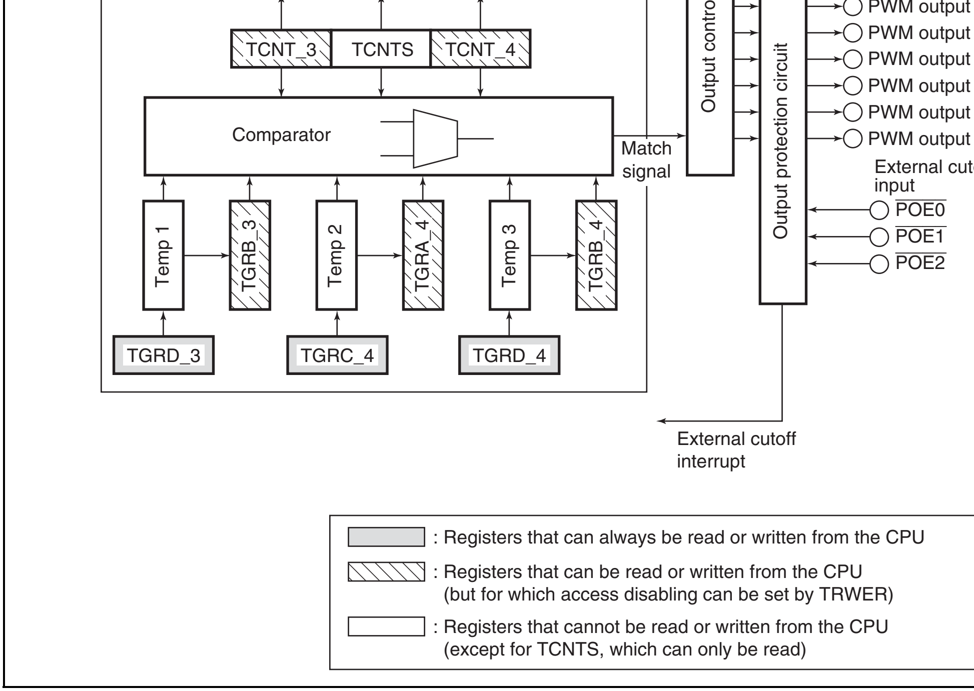
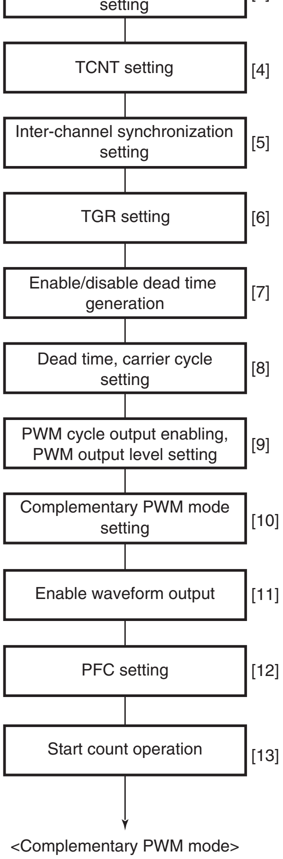

TCNT_3
TCNTS
TCNT_4
Comparator
Match signal
Output contro
Output protection circuit
PWM output
PWM output
PWM output
PWM output
PWM output
PWM output
External cut
input
POE0
POE1
POE2
Temp 1
TGRB_3
Temp 2
TGRA_4
Temp 3
TGRB_4
TGRD_3
TGRC_4
TGRD_4
External cutoff interrupt
▭ : Registers that can always be read or written from the CPU
▨ : Registers that can be read or written from the CPU (but for which access disabling can be set by TRWER)
□ : Registers that cannot be read or written from the CPU (except for TCNTS, which can only be read)

**Figure 10.37 Block Diagram of Channels 3 and 4 in Complementary PWM M**


setting
[3]
TCNT setting
[4]
Inter-channel synchronization setting
[5]
TGR setting
[6]
Enable/disable dead time generation
[7]
Dead time, carrier cycle setting
[8]
PWM cycle output enabling, PWM output level setting
[9]
Complementary PWM mode setting
[10]
Enable waveform output
[11]
PFC setting
[12]
Start count operation
[13]
<Complementary PWM mode>

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 Procedur**

When the CST bit is set to 1, TCNT_3 counts up to the value set in TGRA_3, then sw down-counting when it matches TGRA_3. When the TCNT3 value matches TDDR, t 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, anc switches to down-counting when it matches TCDR. On reaching H'0000, TCNT4 swi 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, dov counting is started, and when TCNTS matches TCDR, the operation switches to up-co 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-cou started, and when TCNTS matches TDDR, the operation switches to down-counting. 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 P is set during the count operation only.

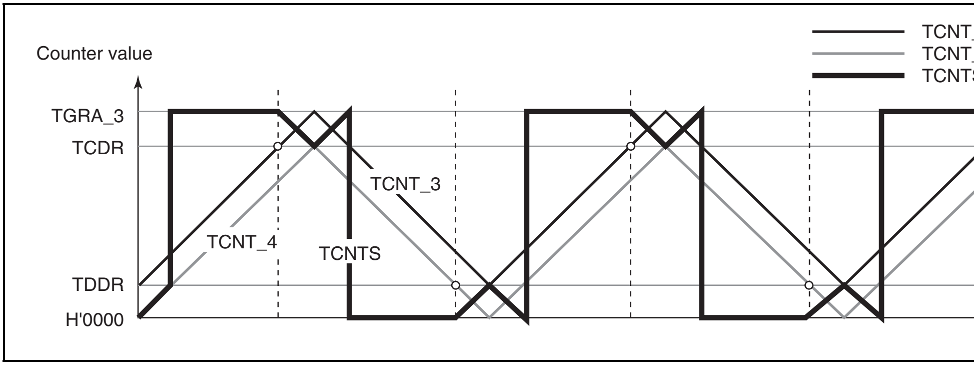


**Figure 10.39 Complementary PWM Mode Counter Operation**

Data in a compare register is changed by writing the new data to the corresponding b 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 registe interval. Data is not transferred to the temporary register in the Tb interval. Data wri buffer register in this interval is transferred to the temporary register at the end of the interval.

The value transferred to a temporary register is transferred to the compare register w TCNTS for which the Tb interval ends matches TGRA_3 when counting up, or H'00 counting down. The timing for transfer from the temporary register to the compare re be selected with bits MD3 to MD0 in the timer mode register (TMDR). Figure 10.40 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 regist performed, the temporary register has the same function as the compare register, and compared with the counter. In this interval, therefore, there are two compare match r for one-phase output, with the compare register containing the pre-change data, and t temporary register containing the new data. In this interval, the three counters—TCN TCNT_4, and TCNTS—and two registers—compare register and temporary register compared, and PWM output controlled accordingly.

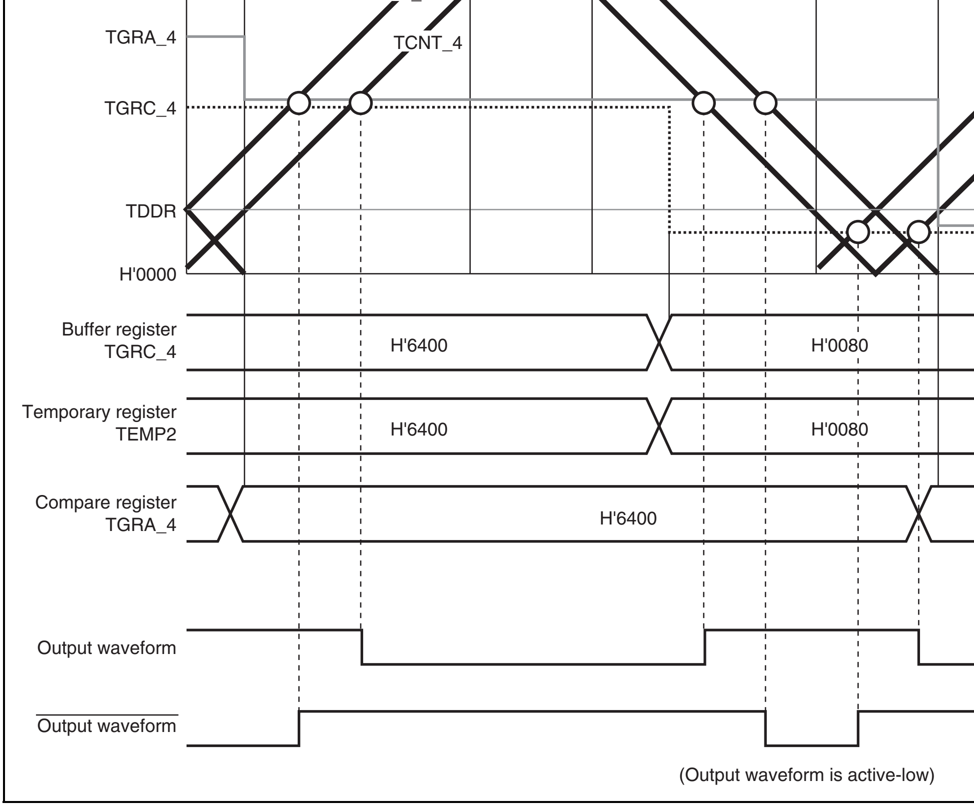


**Figure 10.40 Example of Complementary PWM Mode Operation**

When dead time is not needed, the TDER bit in the timer dead time enable register (
should be cleared to 0, TGRC_3 and TGRA_3 should be set to 1/2 the PWM carrier
and TDDR should be set to 1.

Set the respective initial PWM duty values in buffer registers TGRD_3, TGRC_4, ar
TGRD_4.

The values set in the five buffer registers excluding TDDR are transferred simultane
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 g is disabled by TDER) |
| TDDR | Dead time Td (1 when dead time generation 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 TCB dead time Td set in TDDR. When dead time generation is disabled by TDER, TG must be set to 1/2 the PWM carrier cycle + 1.

The non-overlap time is set in the timer dead time data register (TDDR). The value se TDDR is used as the TCNT_3 counter start value, and creates non-overlap between T and TCNT_4. Complementary PWM mode should be cleared before changing the cor TDDR.

6. Dead Time Suppressing

   Dead time generation is suppressed by clearing the TDER bit in the timer dead time e register (TDER) to 0. TDER can be cleared to 0 only when 0 is written to it after read TDER = 1.

   TGRA_3 and TGRC_3 should be set to 1/2 PWM carrier cycle + 1 and the timer dead data register (TDDR) should be set to 1.

   By the above settings, PWM waveforms without dead time can be obtained. Figure 10 shows an example of operation without dead time.

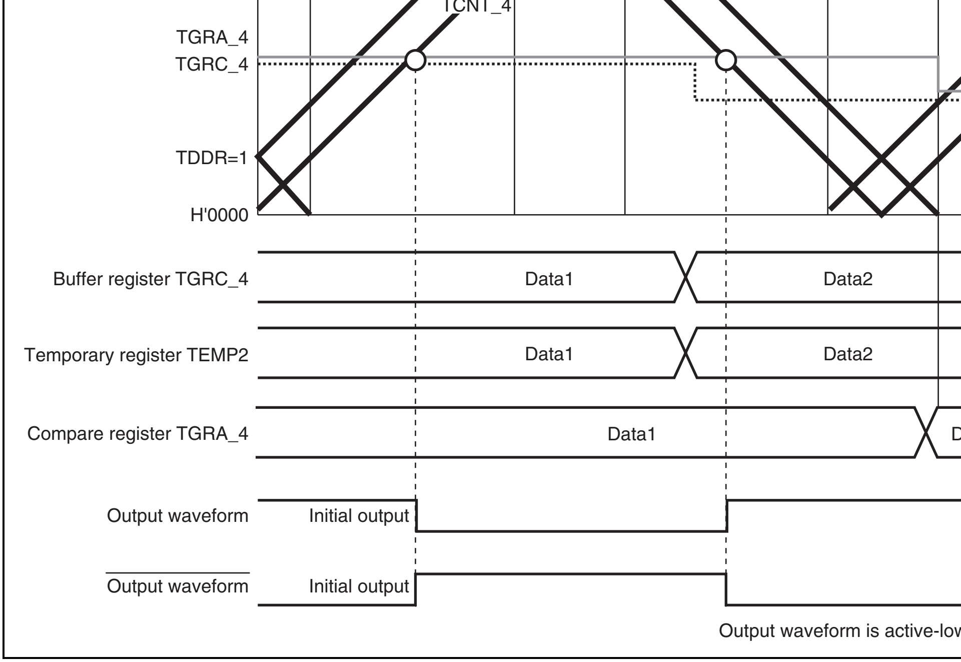


**Figure 10.41 Example of Operation without Dead Time**

and TCDR in accordance with the transfer timing selected with bits MD3 to MD0 in t
mode register (TMDR).

The updated PWM cycle is reflected from the next cycle when the data update is perf
the crest, and from the current cycle when performed in the trough. Figure 10.42 illus
operation when the PWM cycle is updated at the crest.

See the following section, Register Data Updating, for the method of updating the dat
buffer register.

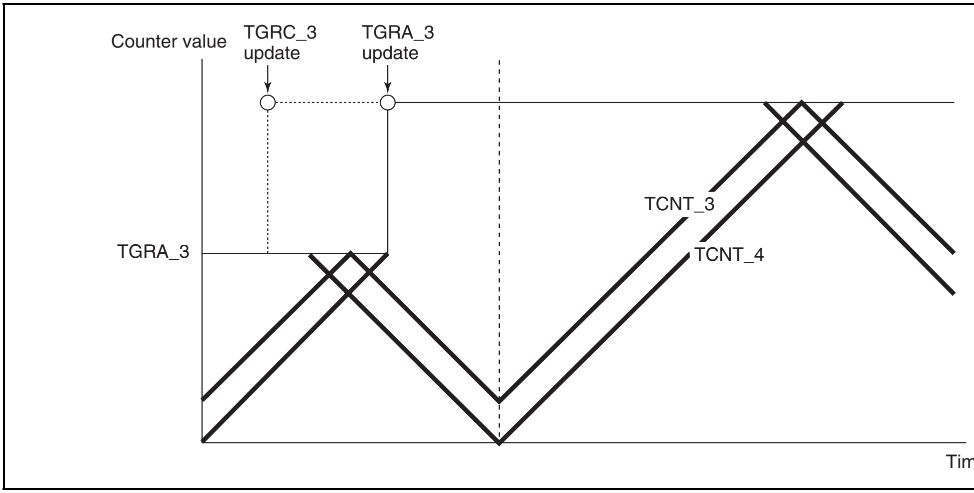


**Figure 10.42 Example of PWM Cycle Updating**

with bits MD3 to MD0 in the timer mode register (TMDR). Figure 10.43 shows an e
data updating in complementary PWM mode. This example shows the mode in whic
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 en
update. Data transfer from the buffer registers to the temporary registers is performe
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 updat
when not updating all five registers, or when updating the TGRD_4 data. In this case
written to TGRD_4 should be the same as the data prior to the write operation.

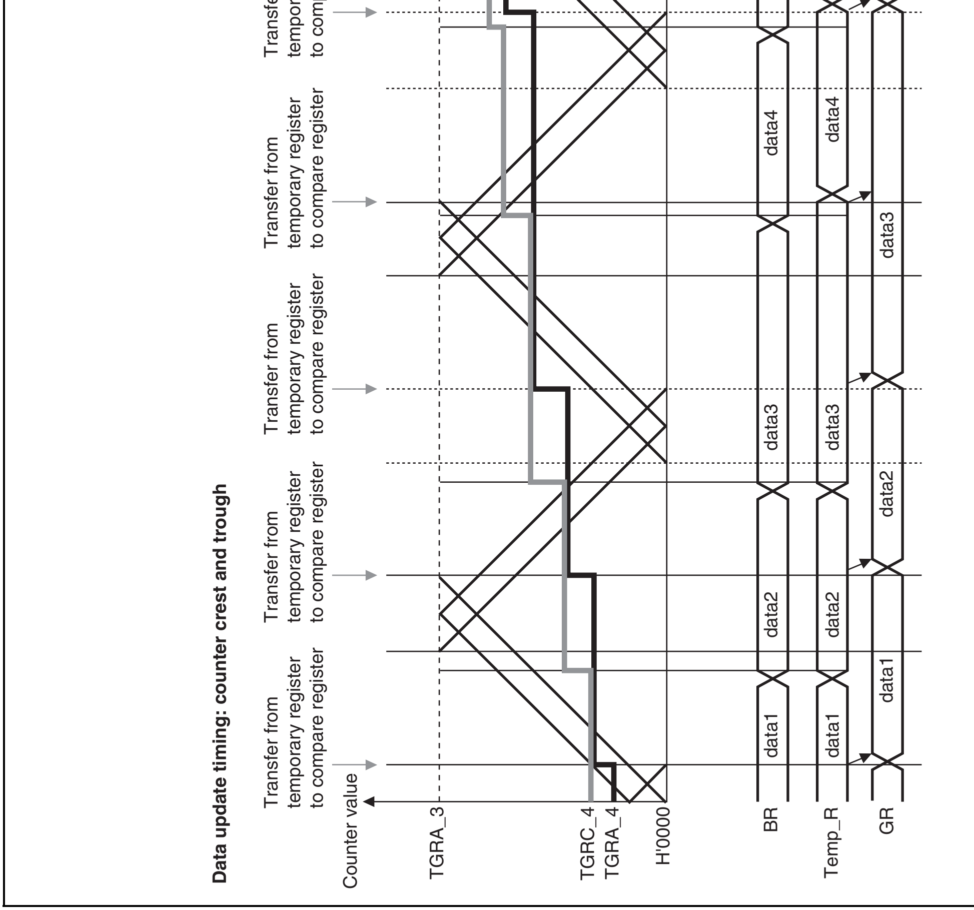

Data update timing: counter crest and trough
Transfer from temporary register to compare register
Transfer from temporary register to compare register
Transfer from temporary register to compare register
Transfer from temporary register to compare register
Transfer from temporary register to compare register
Transfe tempor to comp
Counter value
TGRA_3
TGRC_4
TGRA_4
H'0000
BR
Temp_R
GR
data1
data1
data1
data2
data2
data2
data3
data3
data3
data4
data4

**Figure 10.43 Example of Data Update in Complementary PWM Mode**

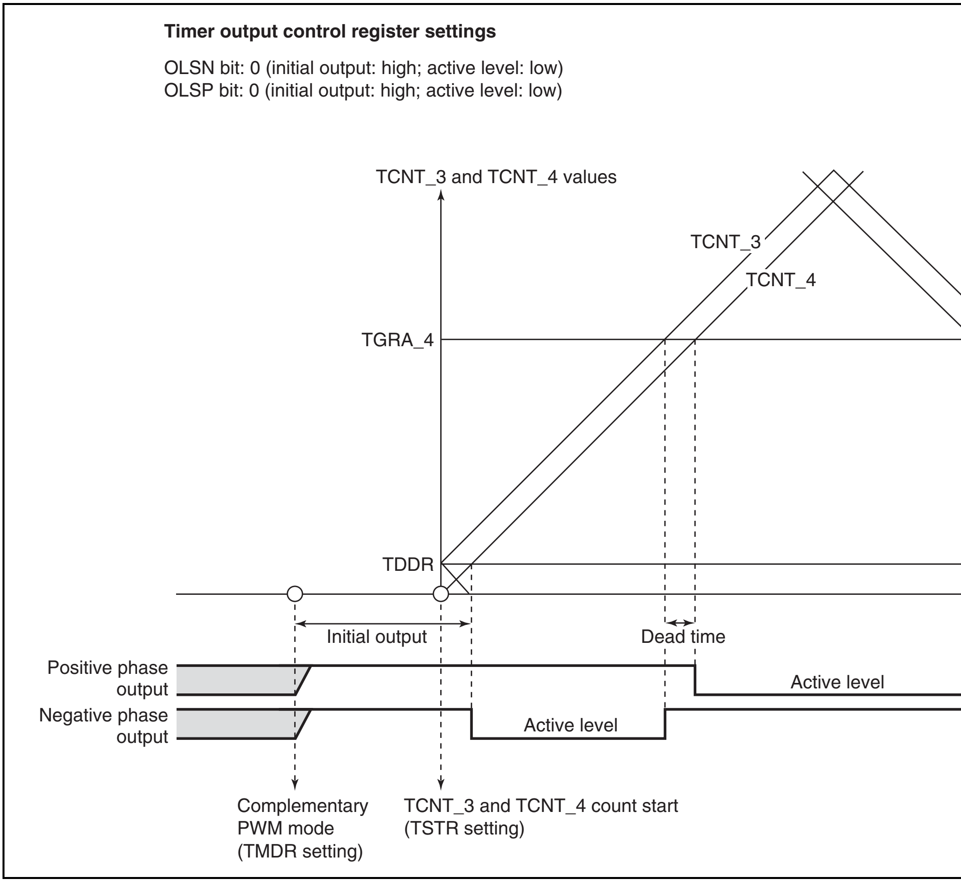

Timer output control register settings
OLSN bit: 0 (initial output: high; active level: low)
OLSP bit: 0 (initial output: high; active level: low)
TCNT_3 and TCNT_4 values
TCNT_3
TCNT_4
TGRA_4
TDDR
Initial output
Dead time
Positive phase output
Negative phase output
Active level
Active level
Complementary PWM mode (TMDR setting)
TCNT_3 and TCNT_4 count start (TSTR setting)

**Figure 10.44 Example of Initial Output in Complementary PWM Mode (**

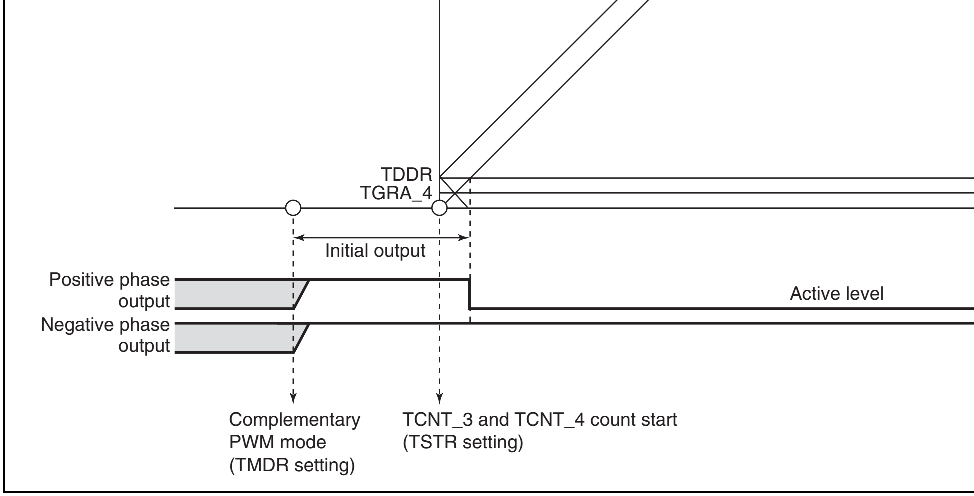

TDDR
TGRA_4
Initial output
Positive phase output
Negative phase output
Active level
Complementary PWM mode (TMDR setting)
TCNT_3 and TCNT_4 count start (TSTR setting)

**Figure 10.45 Example of Initial Output in Complementary PWM Mode (2**

overlap. Figures 10.46 to 10.48 show examples of waveform generation in complem PWM mode.

The positive phase/negative phase off timing is generated by a compare-match with line counter, and the on timing by a compare-match with the dotted-line counter ope a delay of the dead time behind the solid-line counter. In the T1 period, compare-ma turns off the negative phase has the highest priority, and compare-matches occurring are ignored. In the T2 period, compare-match **c** that turns off the positive phase has 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** → as shown in figure 10.46.

If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which negative phase is off is less than twice the dead time, the figure shows the positive p being turned on. If compare-matches deviate from the **c** → **d** → **a'** → **b'** order, since for which the positive phase is off is less than twice the dead time, the figure shows negative phase is not being turned on.

If compare-match **c** occurs first following compare-match **a**, as shown in figure 10.4 compare-match **b** is ignored, and the negative phase is turned off by compare-match because turning off of the positive phase has priority due to the occurrence of compa (positive phase off timing) before compare-match **b** (positive phase on timing) (cons 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 occu **c**, which turns off the positive phase, are ignored. As a result, the negative phase is n on.

Thus, in complementary PWM mode, compare-matches at turn-off timings take prec and turn-on timing compare-matches that occur before a turn-off timing compare-ma ignored.

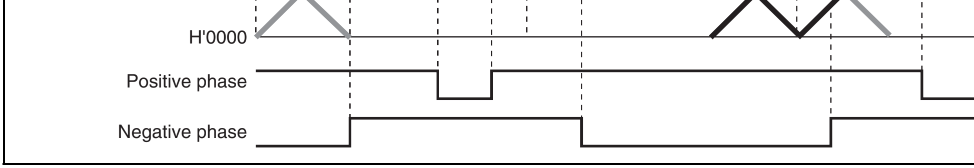


**Figure 10.46   Example of Complementary PWM Mode Waveform Output (**

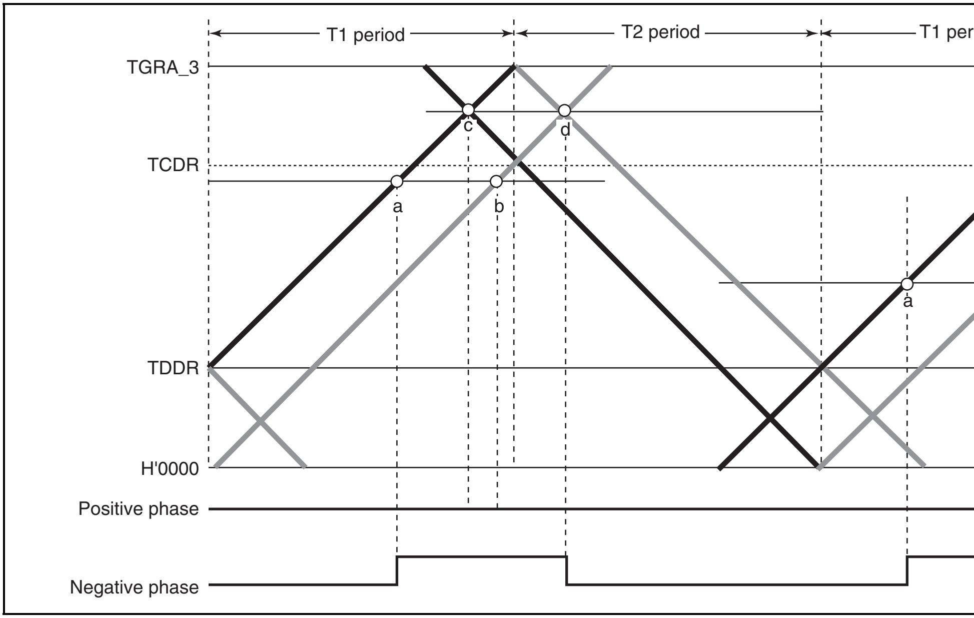


**Figure 10.47   Example of Complementary PWM Mode Waveform Output (**

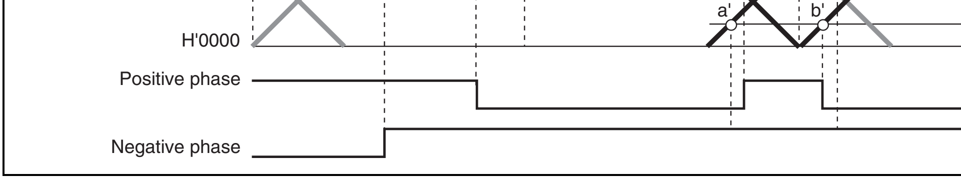


**Figure 10.48 Example of Complementary PWM Mode Waveform Output**

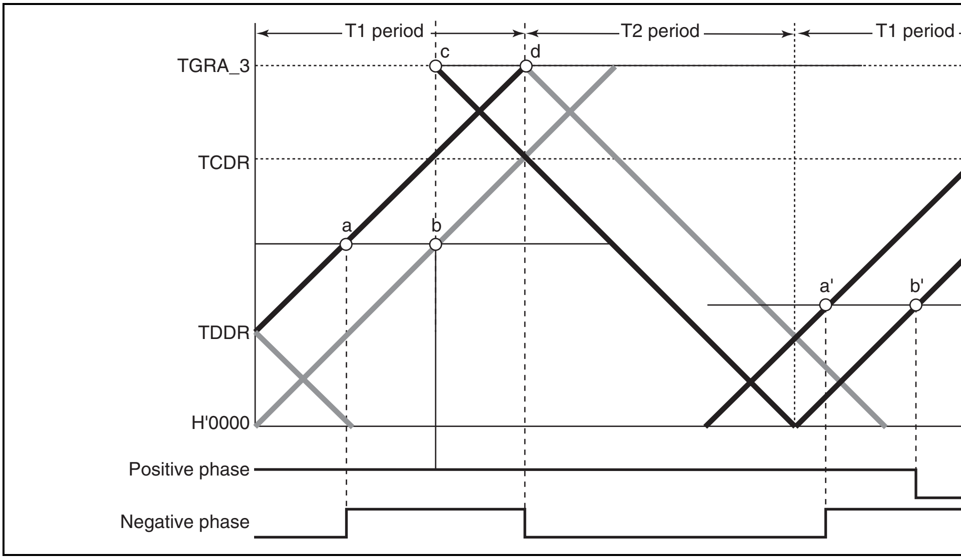


**Figure 10.49 Example of Complementary PWM Mode 0% and 100% Waveform**

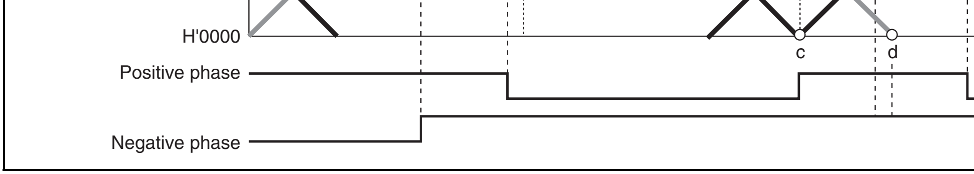


**Figure 10.50 Example of Complementary PWM Mode 0% and 100% Waveform O**

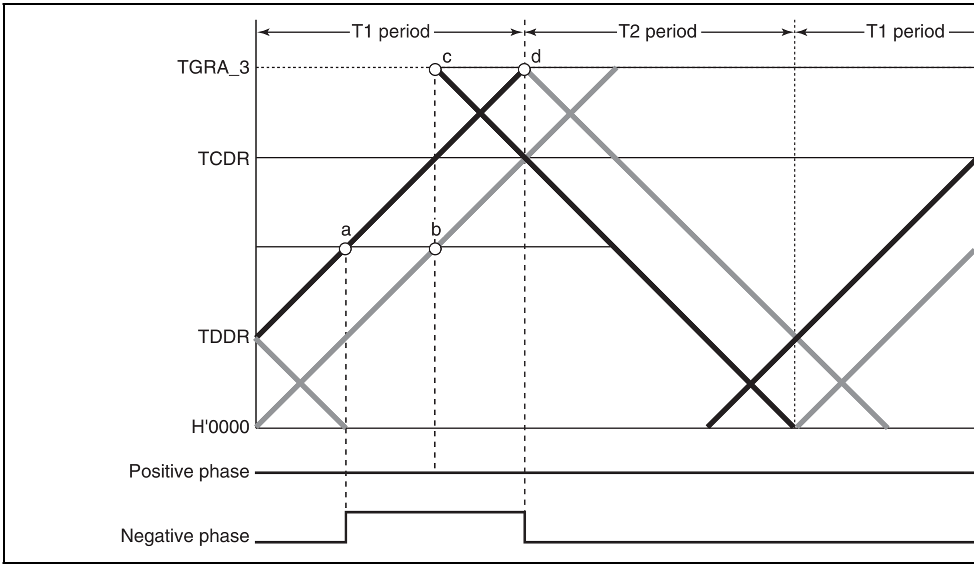


**Figure 10.51 Example of Complementary PWM Mode 0% and 100% Waveform O**

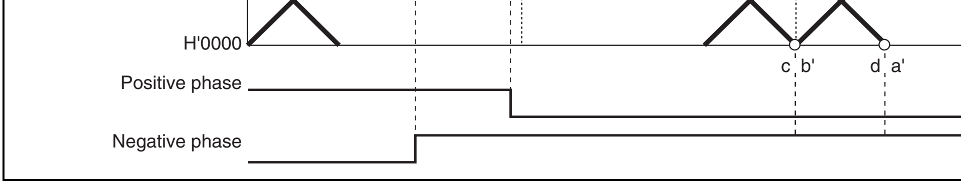


**Figure 10.52 Example of Complementary PWM Mode 0% and 100% Waveform**

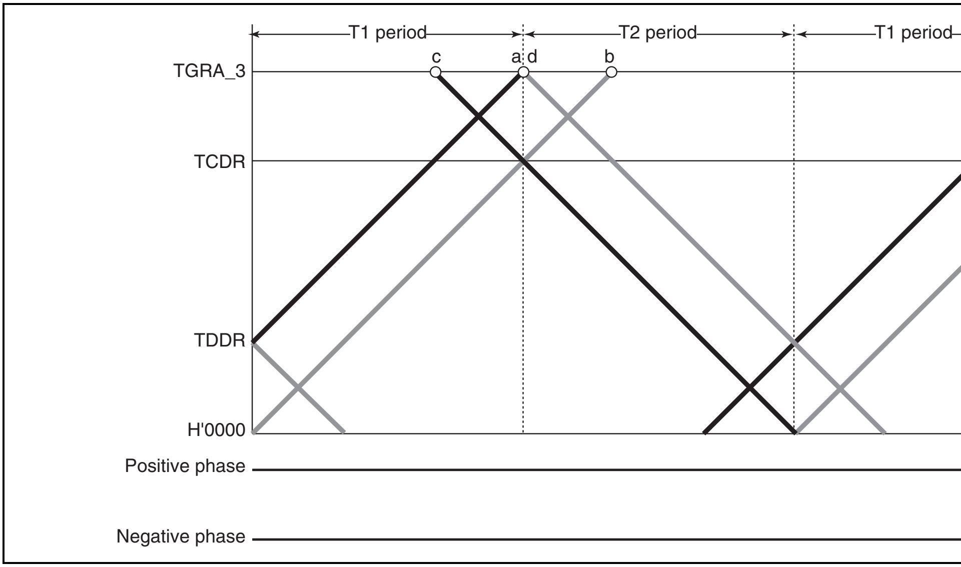


**Figure 10.53 Example of Complementary PWM Mode 0% and 100% Waveform**

12. Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output can be performed in synchronization wi
PWM carrier cycle by setting the PSYE bit to 1 in the timer output control register (T
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 con
match between TCNT4 and H'0000.

The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

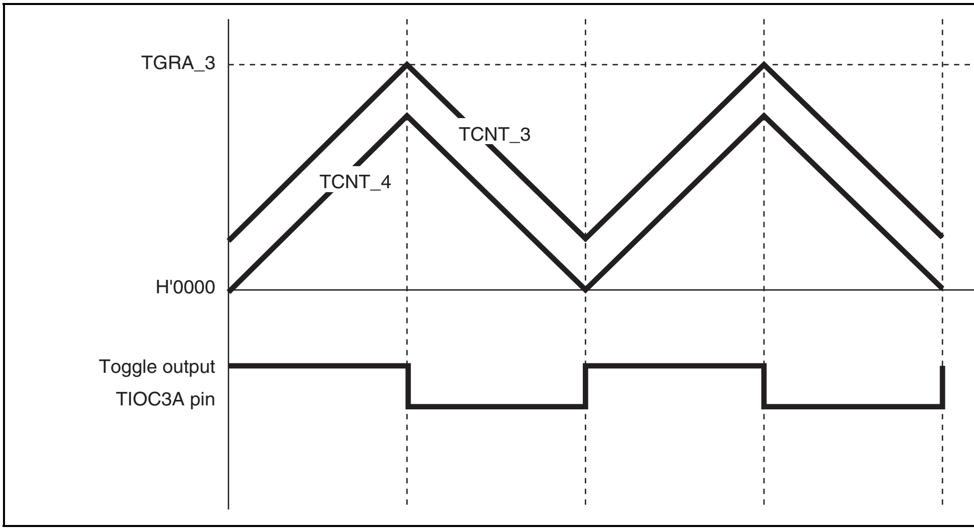


**Figure 10.54   Example of Toggle Output Waveform Synchronized with PWM O**

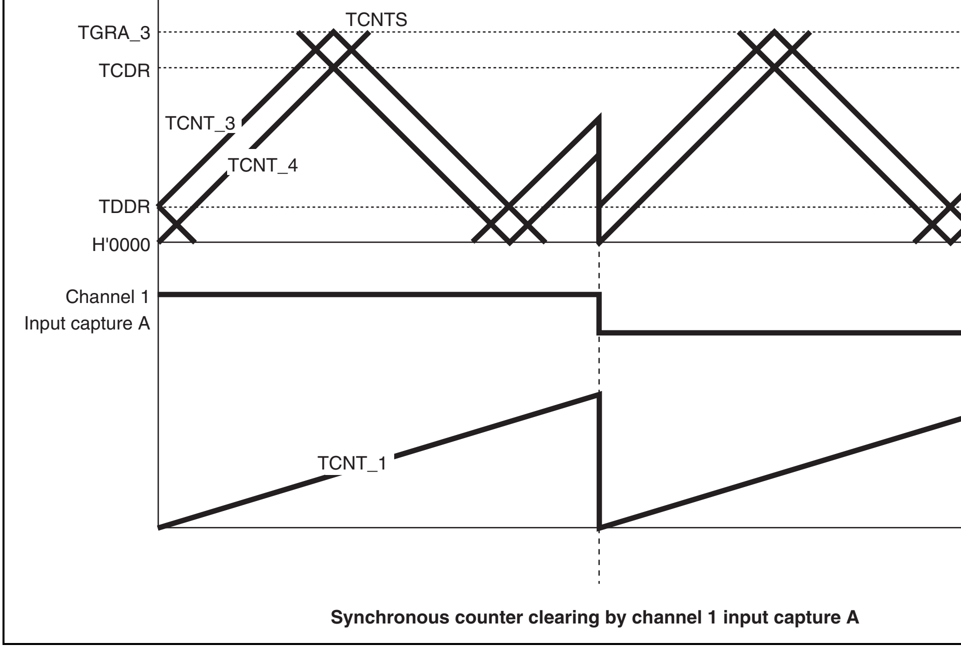


**Figure 10.55 Counter Clearing Synchronized with Another Channel**

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 da register TDDR. If synchronous clearing occurs with the compare registers set to a val than twice the setting of TDDR, the PWM output dead time may be too short (or none 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 Du Synchronous Counter Clearing in Complementary PWM Mode.

This function can be used in both the MTU2 and MTU2S. In the MTU2, synchronous generated in channels 0 to 2 in the MTU2 can cause counter clearing in complementar mode; in the MTU2S, compare match or input capture flag setting in channels 0 to 2 i MTU2 can cause counter clearing.

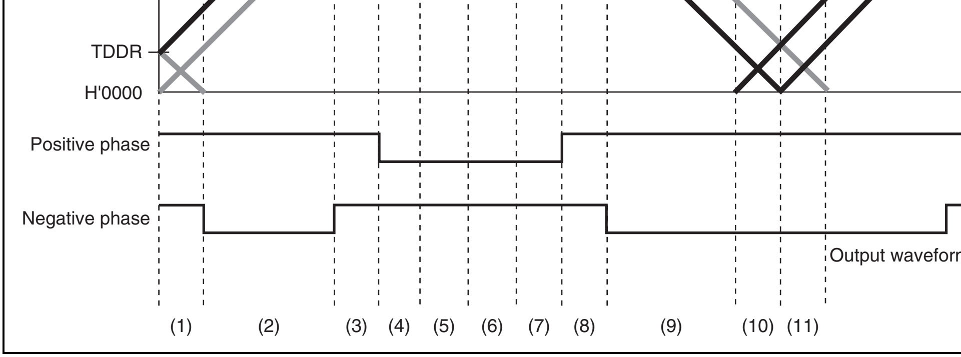
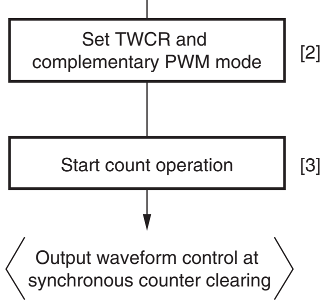


**Figure 10.56 Timing for Synchronous Counter Clearing**


Set TWCR and complementary PWM mode
[2]
Start count operation
[3]
Output waveform control at synchronous counter clearing

to it to suppress initial value output at counter clearing.

[3] Set bits CST3 and CST4 in TSTR to 1 to start count operation.

**Figure 10.57 Example of Procedure for Setting Output Waveform Control at Sync 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 M operates in complementary PWM mode and synchronous counter clearing is gener while the WRE bit in TWCR is set to 1. In the examples shown in figures 10.58 to synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figu respectively.

In the MTU2S, these examples are equivalent to the cases when the MTU2S opera complementary PWM mode and synchronous counter clearing is generated while bit is cleared to 0 and the WRE bit is set to 1 in TWCR.

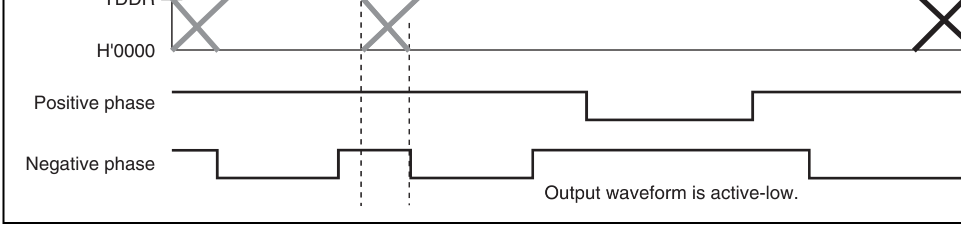


**Figure 10.58 Example of Synchronous Clearing in Dead Time during Up-Cou**
**(Timing (3) in Figure 10.56; Bit WRE of TWCR in MTU2 is 1)**

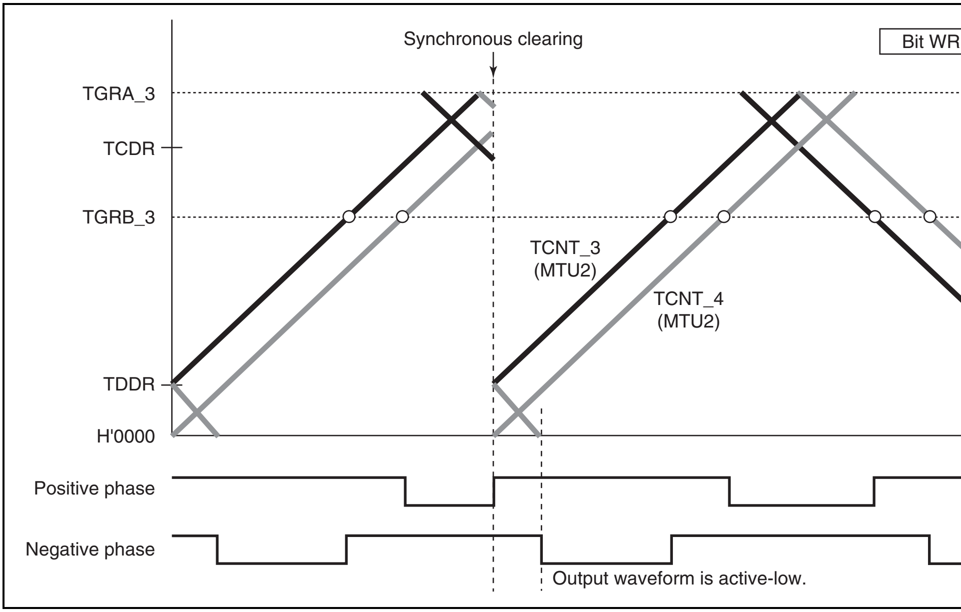


**Figure 10.59 Example of Synchronous Clearing in Interval Tb at Crest**
**(Timing (6) in Figure 10.56; Bit WRE of TWCR in MTU2 is 1)**

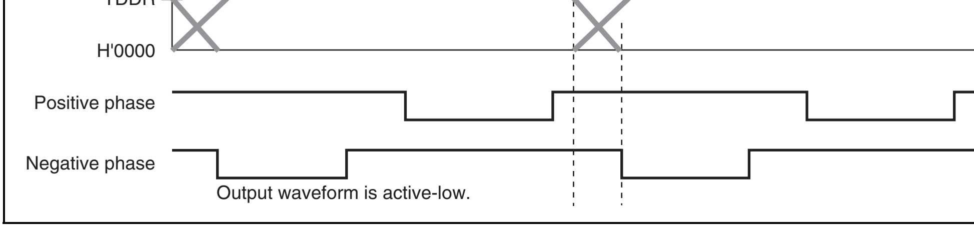


**Figure 10.60 Example of Synchronous Clearing in Dead Time during Down-Cou**
**(Timing (8) in Figure 10.56; Bit WRE of TWCR is 1)**

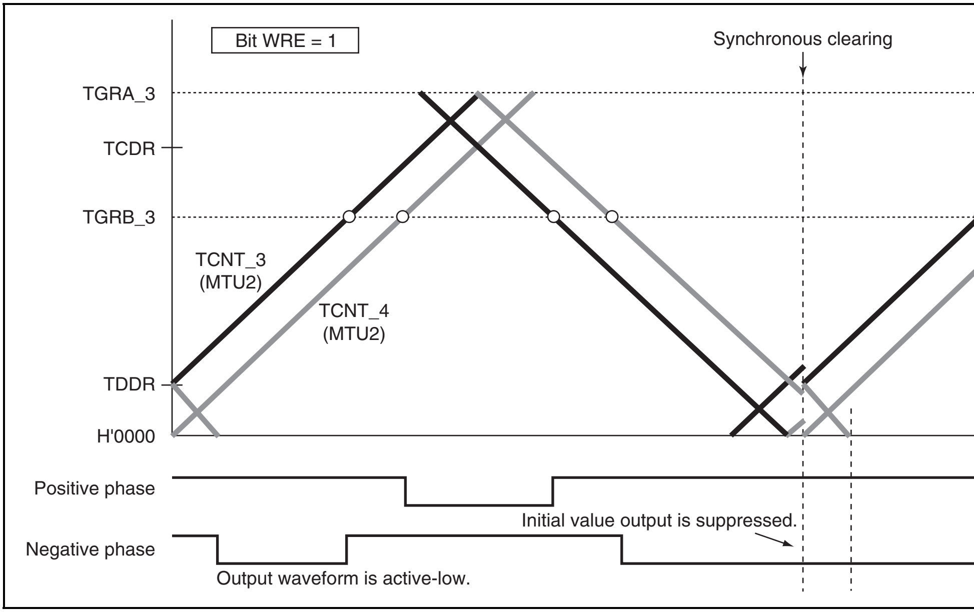


**Figure 10.61 Example of Synchronous Clearing in Interval Tb at Trough**
**(Timing (11) in Figure 10.56; Bit WRE of TWCR is 1)**

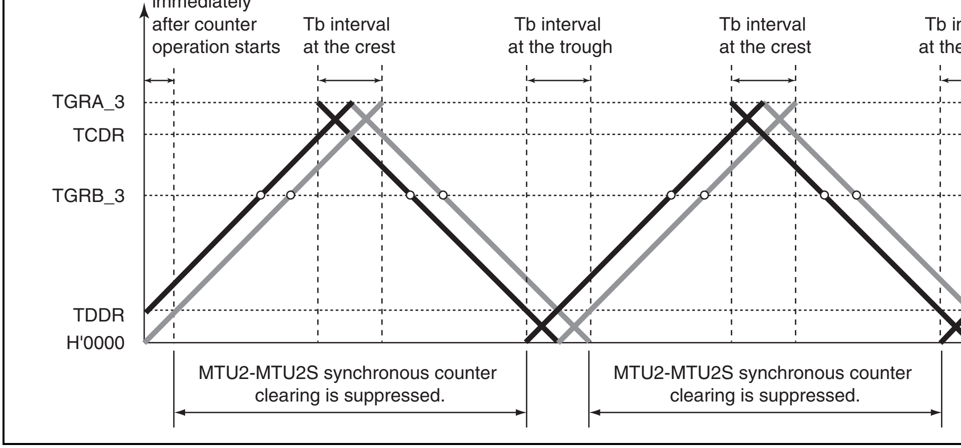
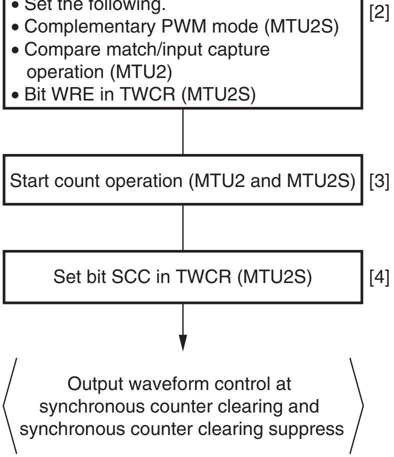


**Figure 10.62 MTU2–MTU2S Synchronous Clearing-Suppressed Interval Specifie Bit in TWCR**


• Set the following.
• Complementary PWM mode (MTU2S)
• Compare match/input capture operation (MTU2)
• Bit WRE in TWCR (MTU2S)
[2]
Start count operation (MTU2 and MTU2S)
[3]
Set bit SCC in TWCR (MTU2S)
[4]
Output waveform control at synchronous counter clearing and synchronous counter clearing suppress

count operation. For MTU2-MTU2S synchronous counte clearing, set bits CST of TSTR in the MTU2 to 1 to start operation in any one of TCNT_0 to TCNT_2.

[4] Read TWCR and then set bit SCC in TWCR to 1 to supp MTU2-MTU2S synchronous counter clearing*. Here, do modify the CCE and WRE bit values in TWCR of the MT MTU2-MTU2S synchronous counter clearing is suppress the intervals shown in figure 10.62.

Note: * The SCC bit value can be modified during counter operation. However, if a synchronous clearing occu when bit SCC is modified from 0 to 1, the synchrono clearing may not be suppressed. If a synchronous clearing occurs when bit SCC is modified from 1 to synchronous clearing may be suppressed.

**Figure 10.63 Example of Procedure for Suppressing MTU2–MTU2S Synchronous 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 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 s figures 10.64 to 10.67, synchronous counter clearing occurs at timing (3), (6), (8), shown in figure 10.56, respectively.

In these examples, the WRE bit in TWCR of the MTU2S is set to 1.

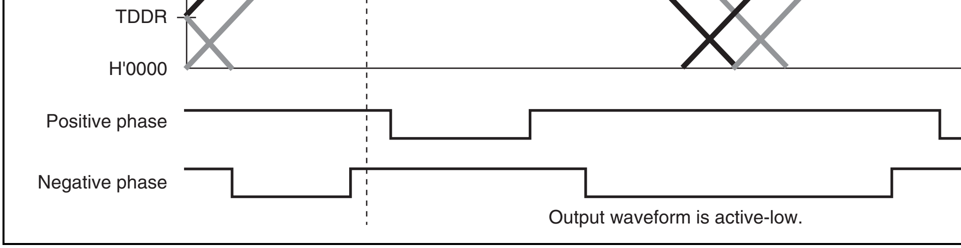


**Figure 10.64 Example of Synchronous Clearing in Dead Time during Up-Cou**
**(Timing (3) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU**

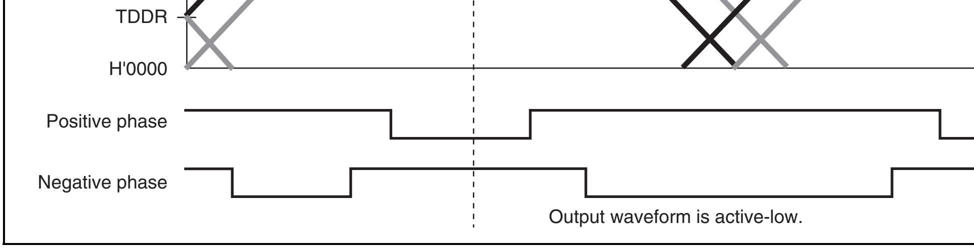


**Figure 10.65 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU**

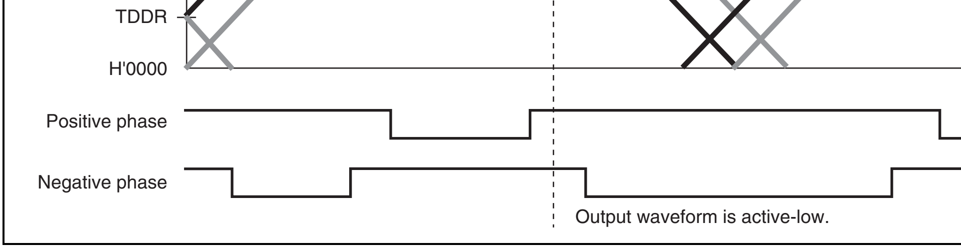


**Figure 10.66 Example of Synchronous Clearing in Dead Time during Down-Co**
**(Timing (8) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU**

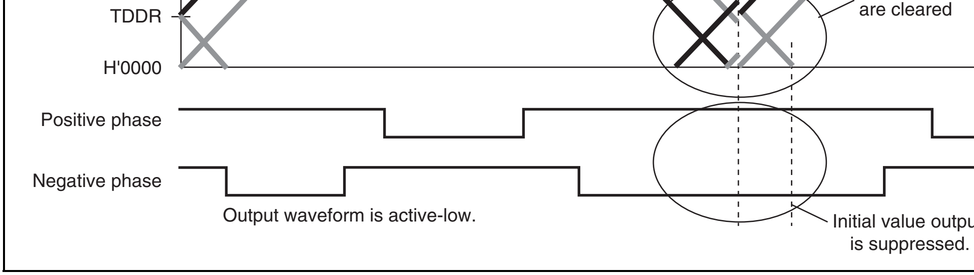


**Figure 10.67 Example of Synchronous Clearing in Interval Tb at Trough**
**(Timing (11) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU**

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.

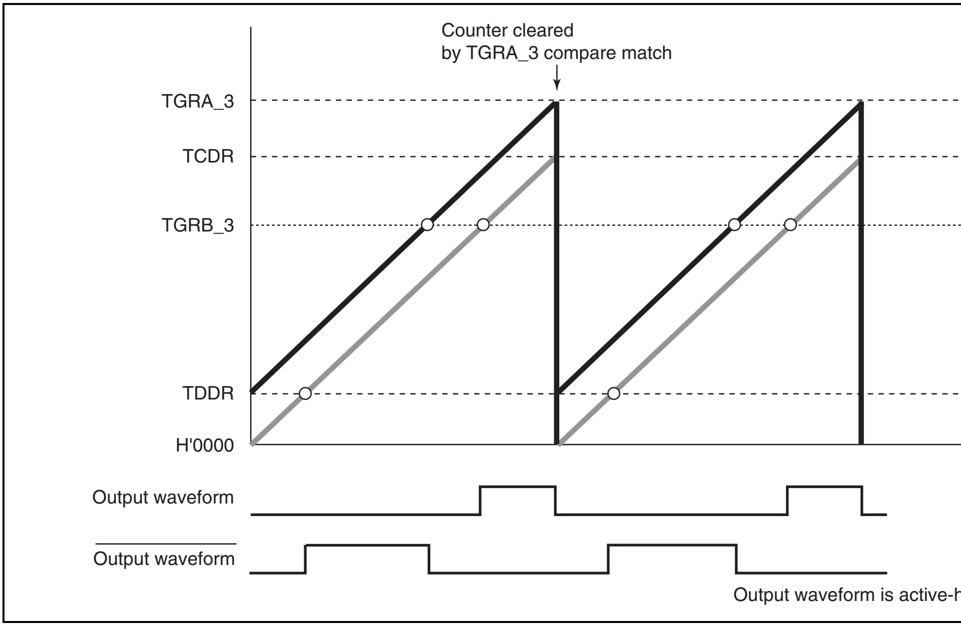
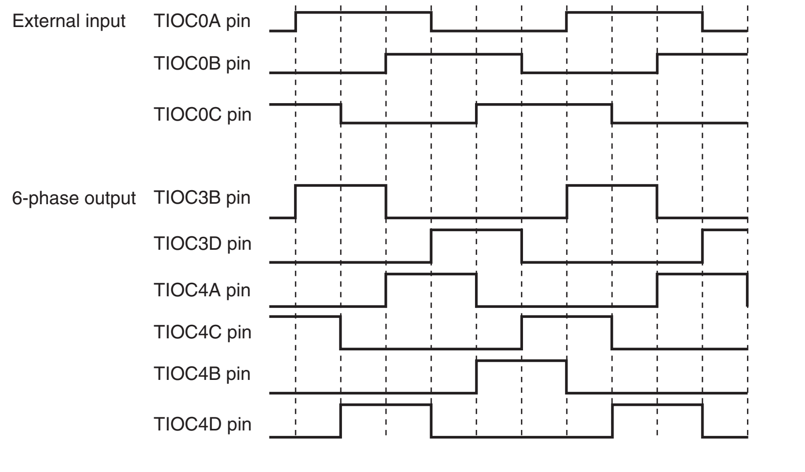


**Figure 10.68 Example of Counter Clearing Operation by TGRA_3 Compare**

is cleared to 0 or set to 1.

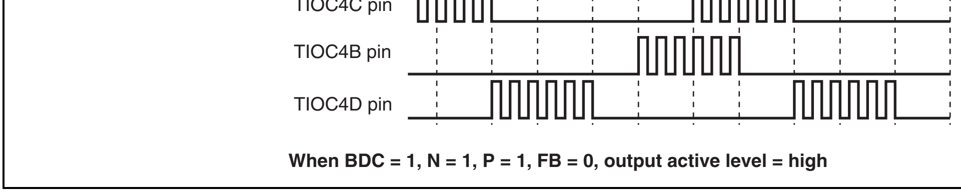
The drive waveforms are output from the complementary PWM mode 6-phase output
With this 6-phase output, in the case of on output, it is possible to use complementary
mode output and perform chopping output by setting the N bit or P bit to 1. When the
P bit is 0, level output is selected.

The 6-phase output active level (on output level) can be set with the OLSN and OLSP
the timer output control register (TOCR) regardless of the setting of the N and P bits.



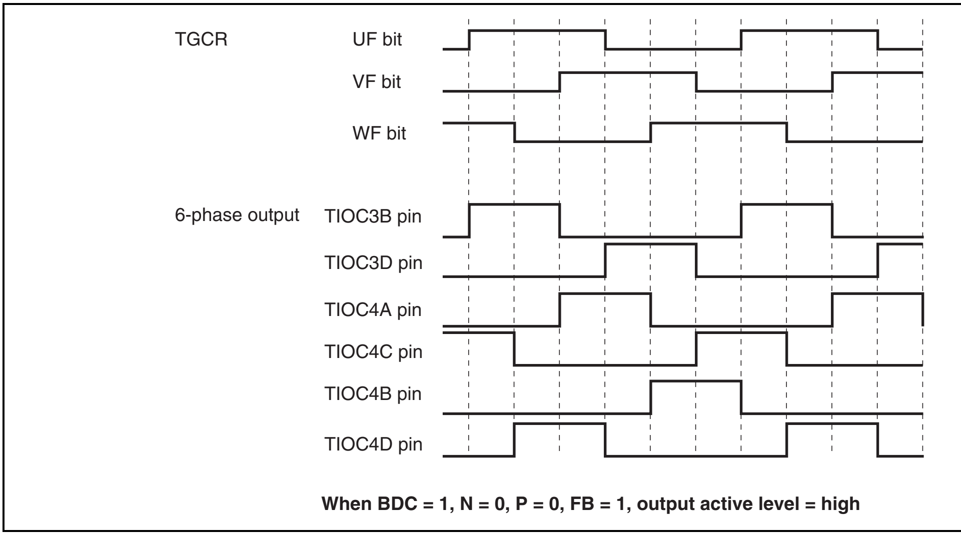
**When BDC = 1, N = 0, P = 0, FB = 0, output active level = high**

**Figure 10.69 Example of Output Phase Switching by External Input (1)**



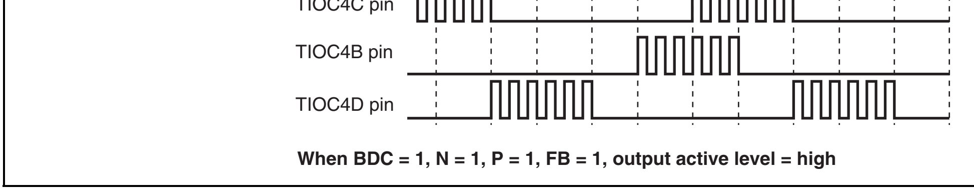
When BDC = 1, N = 1, P = 1, FB = 0, output active level = high

**Figure 10.70 Example of Output Phase Switching by External Input (2)**



When BDC = 1, N = 0, P = 0, FB = 1, output active level = high

**Figure 10.71 Example of Output Phase Switching by Means of UF, VF, WF Bit Se**



When BDC = 1, N = 1, P = 1, FB = 1, output active level = high

**Figure 10.72 Example of Output Phase Switching by Means of UF, VF, WF Bit Set**

18. A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using a T compare-match, TCNT_4 underflow (trough), or compare-match on a channel other t channels 3 and 4.

When start requests using a TGRA_3 compare-match are specified, A/D conversion c started at the crest of the TCNT_3 count.

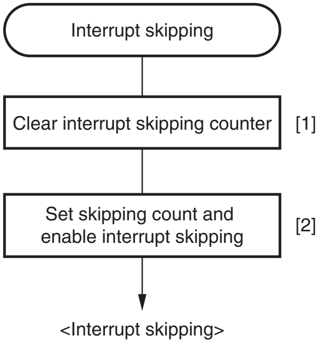
A/D converter start requests can be set by setting the TTGE bit to 1 in the timer interr enable register (TIER). To issue an A/D converter start request at a TCNT_4 underflo (trough), set the TTGE2 bit in TIER_4 to 1.

also be skipped in coordination with interrupt skipping by making settings in the timer
converter request control register (TADCR). For the linkage with the A/D converter star
delaying function, refer to section 10.4.9, A/D Converter Start Request Delaying Functi

The setting of the timer interrupt skipping setting register (TITCR) must be done while t
TGIA_3 and TCIV_4 interrupt requests are disabled by the settings of registers TIER_3
TIER_4 along with under the conditions in which TGFA_3 and TCFV_4 flag settings b
match never occur. Before changing the skipping count, be sure to clear the T3AEN and
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
   10.74 shows the periods during which interrupt skipping count can be changed.



Interrupt skipping
Clear interrupt skipping counter
[1]
Set skipping count and enable interrupt skipping
[2]
<Interrupt skipping>

[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 Procedur**

**Figure 10.74   Periods during which Interrupt Skipping Count can be Chang**

2. Example of Interrupt Skipping Operation

   Figure 10.75 shows an example of TGIA_3 interrupt skipping in which the interrupt s count is set to three by the 3ACOR bit and the T3AEN bit is set to 1 in the timer inter skipping set register (TITCR).

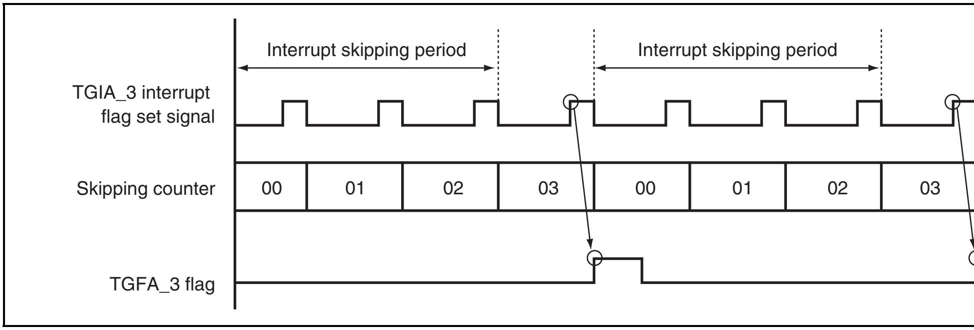
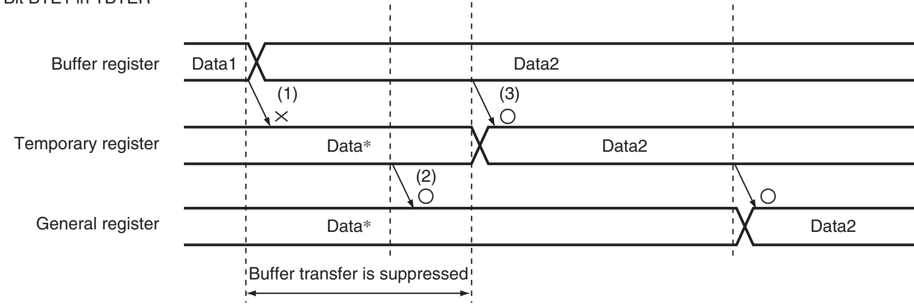

Interrupt skipping period
Interrupt skipping period
TGIA_3 interrupt flag set signal
Skipping counter
00
01
02
03
00
01
02
03
TGFA_3 flag

**Figure 10.75   Example of Interrupt Skipping Operation**

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit the timer interrupt skipping set register (TITCR). Figure 10.78 shows the relationshi 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 in skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3A 4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (T 0). If buffer transfer is linked with interrupt skipping while interrupt skipping is buffer transfer is never performed.

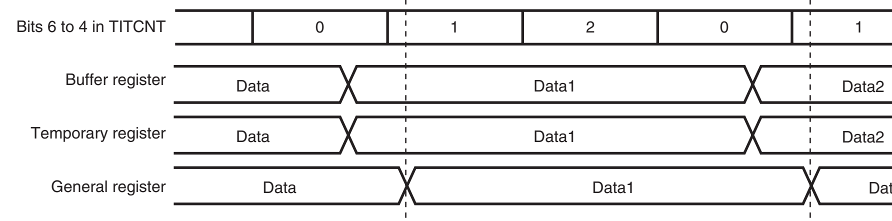


[Legend]

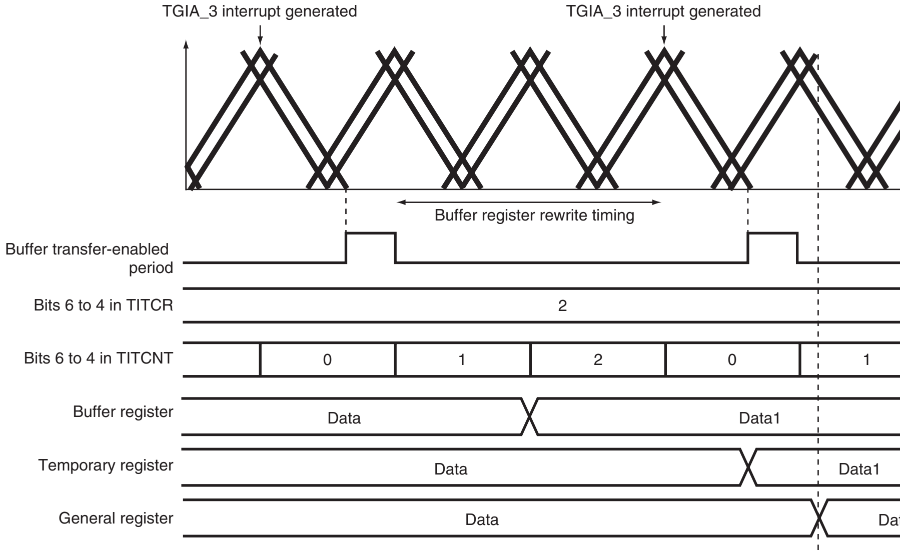
(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).
(2) Data is transferred from the temporary register to the general register even in the buffer transfer-disabled period.
(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)**

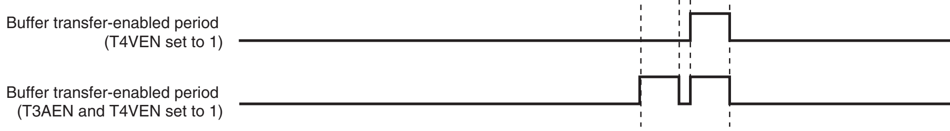


(2) When buffer register is rewritten after one carrier cycle has elapsed after the TGIA_3 interrupt



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 Inte Skipping (BTE1 = 1 and BTE0 = 0)**


Buffer transfer-enabled period
(T4VEN set to 1)
Buffer transfer-enabled period
(T3AEN and T4VEN set to 1)

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 thre
T3AEN and T4VEN are set to 1.

**Figure 10.78 Relationship between Bits T3AEN and T4VEN in Timer Interrupt S 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
   CPU can be enabled or disabled for the mode registers, control registers, compare reg
   and counters used in complementary PWM mode by means of the RWE bit in the tim
   read/write enable register (TRWER). The applicable registers are some (21 in total) o
   registers in channels 3 and 4 shown in the following:

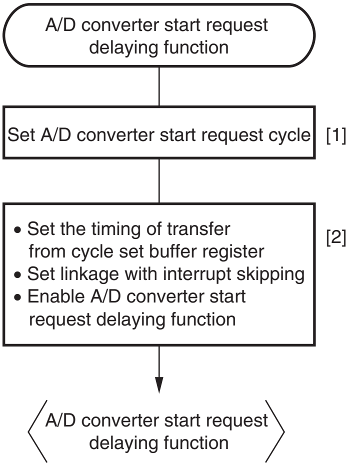
   — TCR_3 and TCR_4, TMDR_3 and TMDR_4, TIORH_3 and TIORH_4, TIORL_
   TIORL_4, TIER_3 and TIER_4, TCNT_3 and TCNT_4, TGRA_3 and TGRA_4,
   and TGRB_4, TOER, TOCR, TGCR, TCDR, and TDDR.

   This function enables miswriting due to CPU runaway to be prevented by disabling C
   access to the mode registers, control registers, and counters. When the applicable regi
   read in the access-disabled state, undefined values are returned. Writing to these regis
   ignored.

A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination wi interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bi TADCR.

1. Example of Procedure for Specifying A/D Converter Start Request Delaying Functior

   Figure 10.79 shows an example of procedure for specifying the A/D converter start re delaying function.


A/D converter start request delaying function
Set A/D converter start request cycle
[1]
• Set the timing of transfer from cycle set buffer register
• Set linkage with interrupt skipping
• Enable A/D converter start request delaying function
[2]
A/D converter start request delaying function

[1] Set the cycle in the timer A/D converter start request cycl buffer register (TADCOBRA_4 or TADCOBRB_4) and tin A/D converter start request cycle register (TADCORA_4 TADCORB_4). (The same initial value must be specified 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 registe
- Specify whether to link with interrupt skipping through b ITA3AE, ITA4VE, ITB3AE, and ITB4VE.
- Use bits TU4AE, DT4AE, UT4BE, and DT4BE to enabl 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 mo 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**

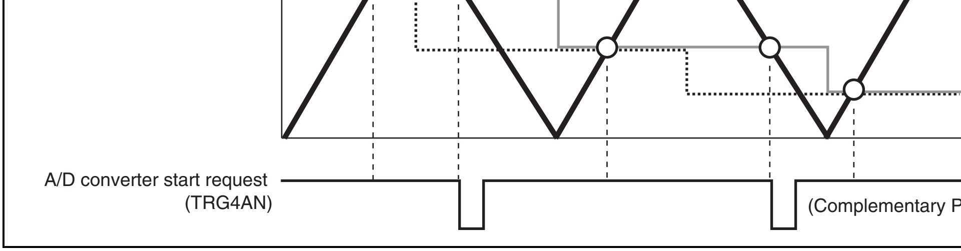


**Figure 10.80 Basic Example of A/D Converter Start Request Signal (TRG4AN) O**

3. Buffer Transfer

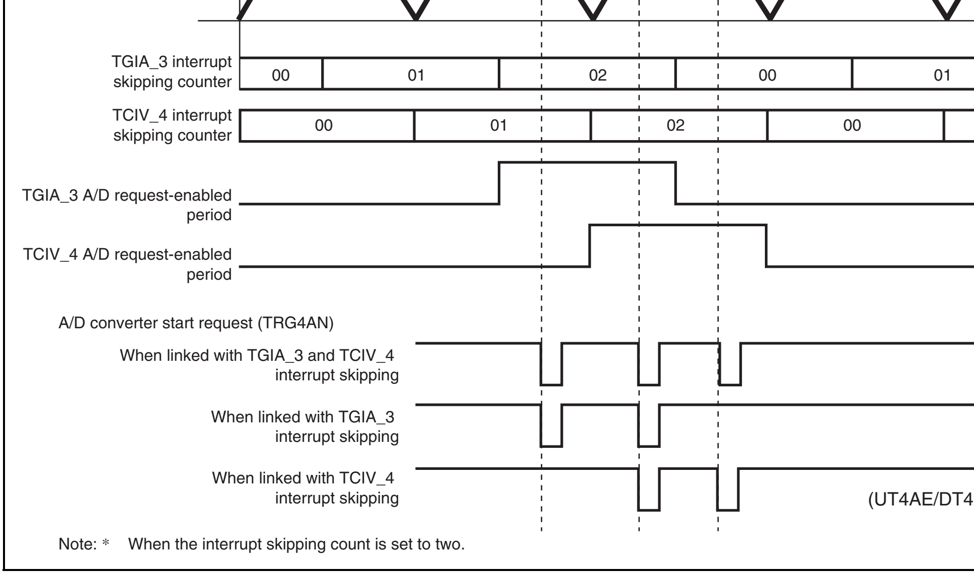
   The data in the timer A/D converter start request cycle set registers (TADCORA_4 a TADCORB_4) is updated by writing data to the timer A/D converter start request cy buffer registers (TADCOBRA_4 and TADCOBRB_4). Data is transferred from the registers to the respective cycle set registers at the timing selected with the BF1 and 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 interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4V the timer A/D converter start request control register (TADCR).

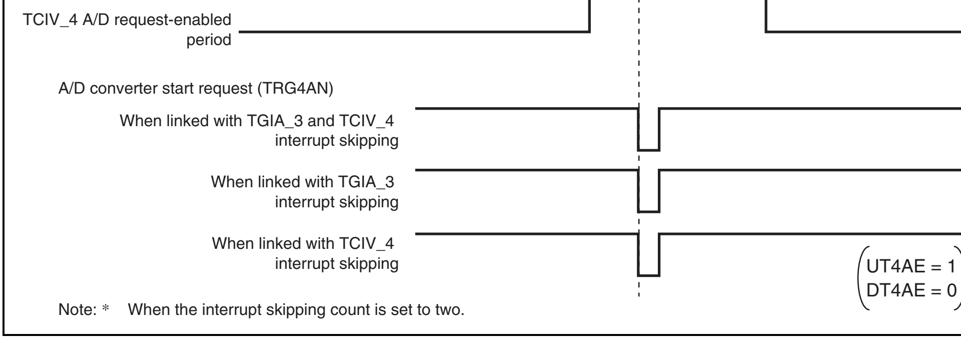
   Figure 10.81 shows an example of A/D converter start request signal (TRG4AN) ope when TRG4AN output is enabled during TCNT_4 up-counting and down-counting a converter start requests are linked with interrupt skipping.

   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 co start requests are linked with interrupt skipping.


TGIA_3 interrupt skipping counter
00
01
02
00
01
TCIV_4 interrupt skipping counter
00
01
02
00
TGIA_3 A/D request-enabled period
TCIV_4 A/D request-enabled period
A/D converter start request (TRG4AN)
When linked with TGIA_3 and TCIV_4 interrupt skipping
When linked with TGIA_3 interrupt skipping
When linked with TCIV_4 interrupt skipping
(UT4AE/DT4

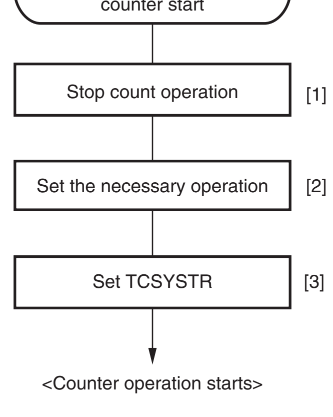
Note: * When the interrupt skipping count is set to two.

**Figure 10.81 Example of A/D Converter Start Request Signal (TRG4AN) Operatio with Interrupt Skipping**


TCIV_4 A/D request-enabled period
A/D converter start request (TRG4AN)
When linked with TGIA_3 and TCIV_4 interrupt skipping
When linked with TGIA_3 interrupt skipping
When linked with TCIV_4 interrupt skipping
UT4AE = 1
DT4AE = 0

Note: * When the interrupt skipping count is set to two.

**Figure 10.82 Example of A/D Converter Start Request Signal (TRG4AN) Operati with Interrupt Skipping**



[2] Specify necessary operation with appropriate registers such 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 operati counter is cleared to 0, the counter will not stop. To st 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 appropria settings in TCYSTR according to the TSTR setting for the respective mode. For details, refer to section 10.4 Reset-Synchronized PWM Mode, and section 10.4.8, Complementary PWM Mode.

Figure 10.83 Example of Synchronous Counter Start Setting Procedure

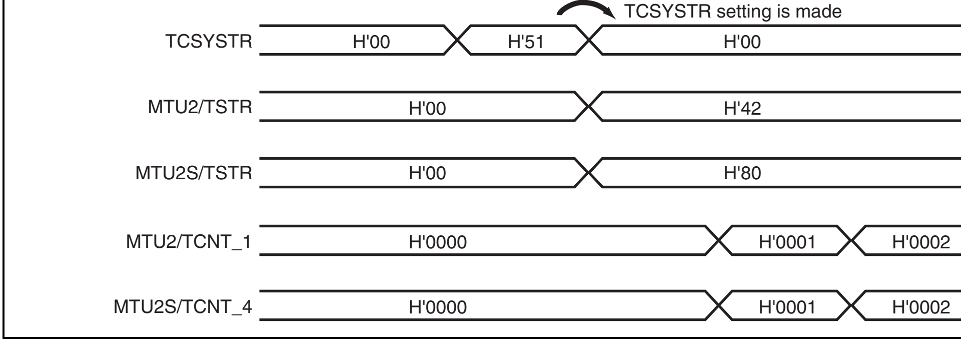


**Figure 10.84 (1) Example of Synchronous Counter Start Operation (MTU2-to-**
**Clock Frequency Ratio = 1:1)**

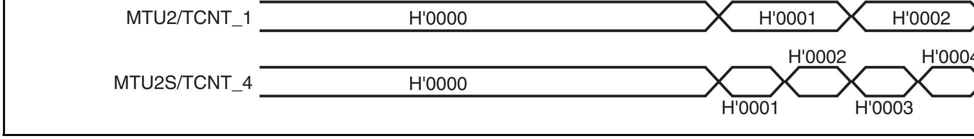


**Figure 10.84 (2) Example of Synchronous Counter Start Operation (MTU2-to-M Clock Frequency Ratio = 1:2)**

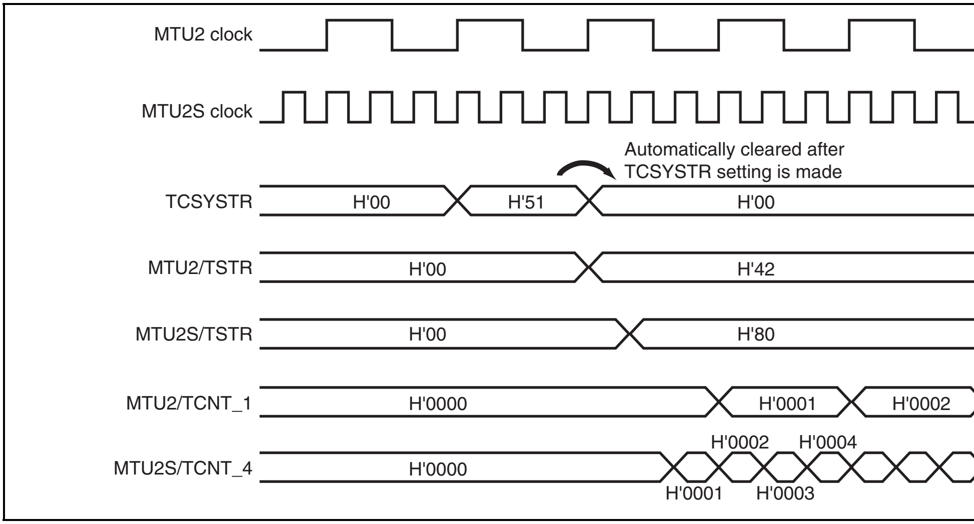


**Figure 10.84 (3) Example of Synchronous Counter Start Operation (MTU2-to-M Clock Frequency Ratio = 1:3)**

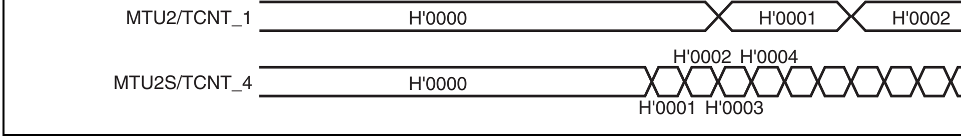
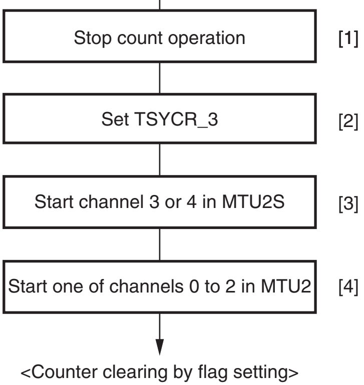


**Figure 10.84 (4) Example of Synchronous Counter Start Operation (MTU2-to-I Clock Frequency Ratio = 1:4)**


Stop count operation
[1]
Set TSYCR_3
[2]
Start channel 3 or 4 in MTU2S
[3]
Start one of channels 0 to 2 in MTU2
[4]
<Counter clearing by flag setting>

[2] Use TSYCR_3 in the MTU2S to specify the flag setting sou to be used for the TCNT_3 and TCNT_4 clearing source.

[3] Start TCNT_3 or TCNT_4 in the MTU2S.

[4] Start TCNT_0, TCNT_1, or TCNT_2 in the MTU2.

Note: The TSYCR_3 setting is ignored while the counter is stopped. The setting becomes valid after TCNT_3 or TCNT4 is started.

**Figure 10.85 Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source**

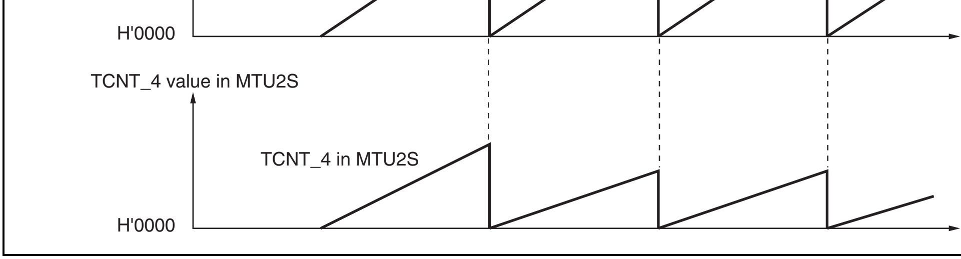


**Figure 10.86 (1) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (1)**

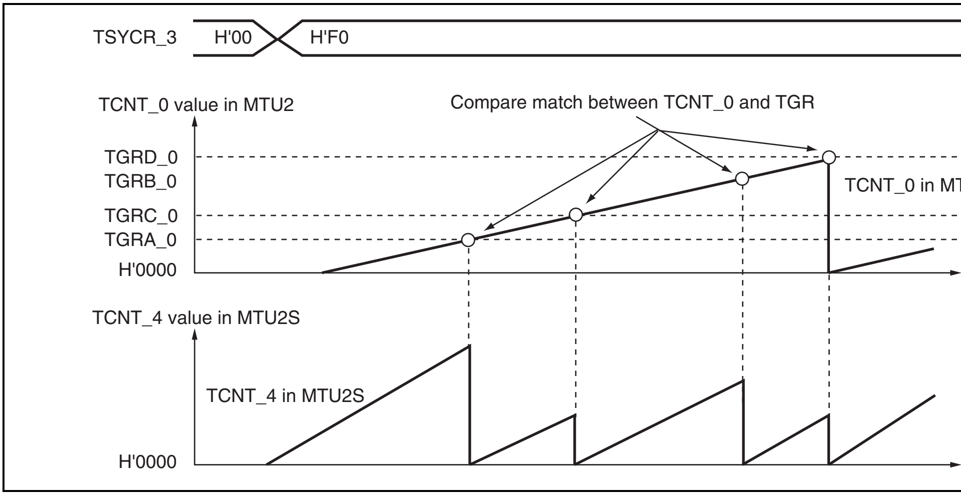
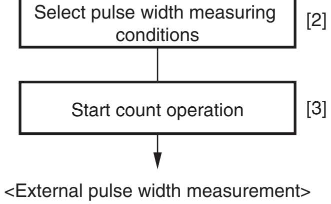


**Figure 10.86 (2) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (2)**


Select pulse width measuring conditions
[2]
Start count operation
[3]
<External pulse width measurement>

Notes: 1. Do not set bits CMPCLR5U, CMPCLR5V, or CMPCLR5W in TCNTCMPCLR to 1.
2. Do not set bits TGIE5U, TGIE5V, or TGIE5W in TIER_5 to 1.
3. The value in TCNT is not captured in TGR.

**Figure 10.87 Example of External Pulse Width Measurement Setting Procedu**

### Example of External Pulse Width Measurement:

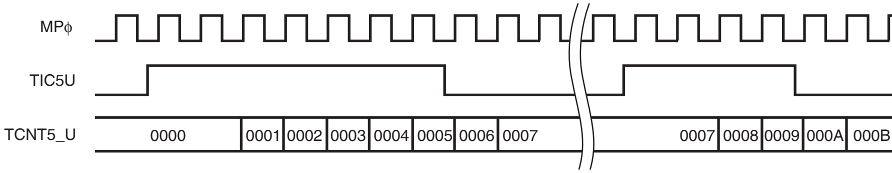


**Figure 10.88 Example of External Pulse Width Measurement (Measuring High Pulse Width)**

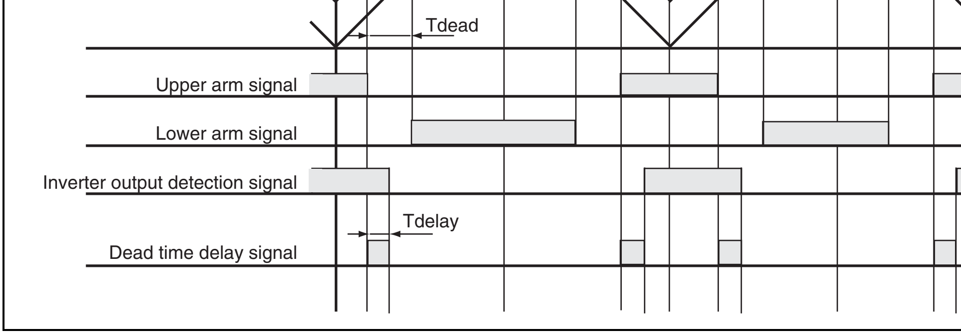
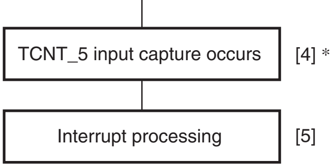


Figure 10.89 Delay in Dead Time in Complementary PWM Operation



the TCNT_5 value is captured in TGR_5.

[5] For U-phase dead time compensation, when an interrupt 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_ and TGRD_4, respectively, in the same way as for U-pha 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 conditio specified in TIOR, do not enable interrupt requests i TIER_5.

**Figure 10.90 Example of Dead Time Compensation Setting Procedure**

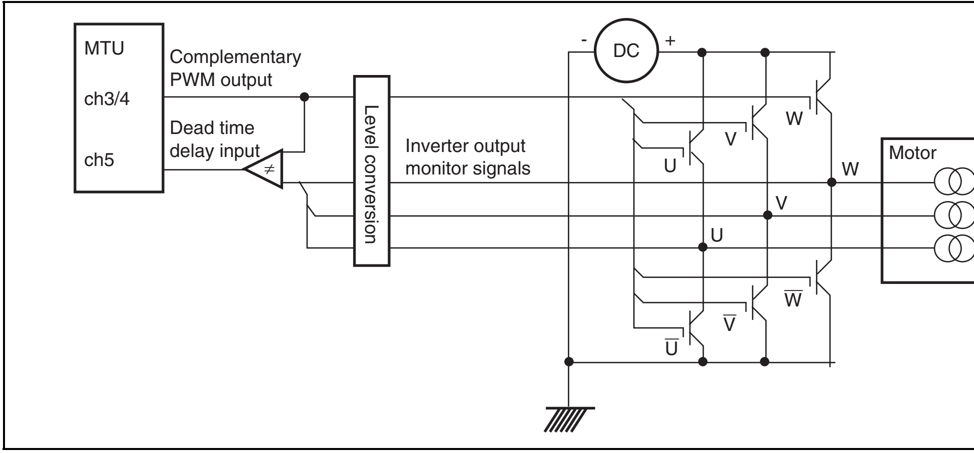


**Figure 10.91 Example of Motor Control Circuit Configuration**

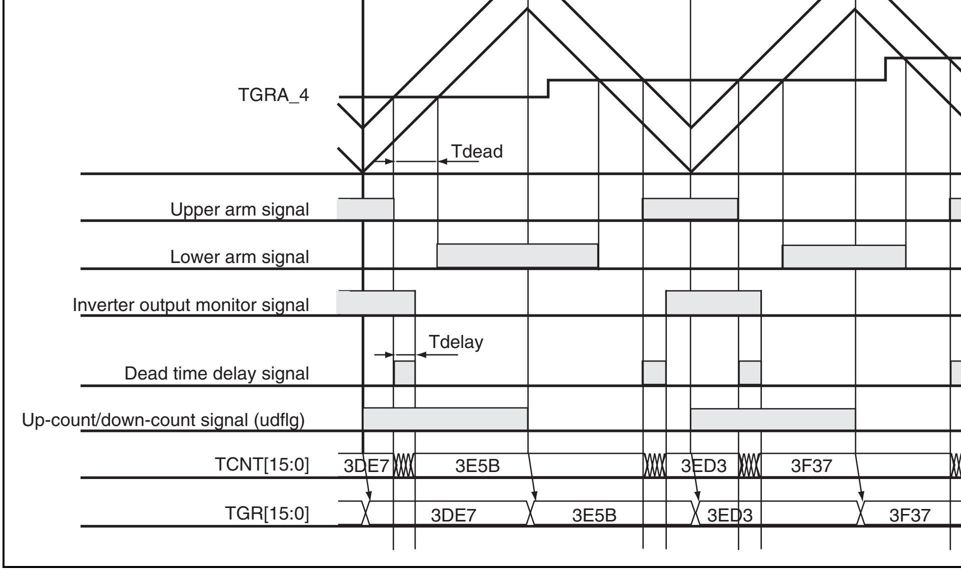


**Figure 10.92 TCNT Capturing at Crest and/or Trough in Complementary PWM**

Relative channel priorities can be changed by the interrupt controller, however the priorit within a channel is fixed. For details, see section 6, Interrupt Controller (INTC).

Table 10.57 lists the MTU2 interrupt sources.

| | | | | |
|---|---|---|---|---|
| | 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 |

Note: This table shows the initial state immediately after a reset. The relative channel p can be changed by the interrupt controller.

**Underflow Interrupt:** An interrupt is requested if the TCIEU bit in TIER is set to 1 whe TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The int request is cleared by clearing the TCFU flag to 0. The MTU2 has two underflow interrup 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 in each channel or the overflow interrupt in channel 4. For details, see section 8, Data Trans Controller (DTC).

A total of 20 MTU2 input capture/compare match interrupts and overflow interrupts can b as DTC activation sources, four each for channels 0 and 3, two each for channels 1 and 2, channel 4, and three for channel 5.

A/D converter start request signal TRGAN is issued to the A/D converter under either o
following conditions.

- When the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/c
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 complemen
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
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
converter can be activated by generating A/D converter start request signal TRG0N whe
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 betwe
TCNT_0 and TGRE_0 in channel 0 while the TTGE2 bit in TIER2_0 is set to 1, A/D co
start request TGR0N is issued to the A/D converter. If A/D converter start signal TGR0
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
converter can be activated by generating A/D converter start request signal TRG4AN or
when the TCNT_4 count matches the TADCORA or TADCORB value if the TAD4AE
TAD4BE bit in the A/D converter start request control register (TADCR) is set to 1. For
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 sele
trigger in the A/D converter when TRG4AN is generated or if TRG4BN from the MTU
selected as the trigger in the A/D converter when TRG4BN is generated.

| | | |
|---|---|---|
| | complementary PWM mode | |
| TGRE_0 and TCNT_0 | Compare match | TRG0N |
| TADCORA and TCNT_4 | | TRG4AN |
| TADCORB and TCNT_4 | | TRG4BN |

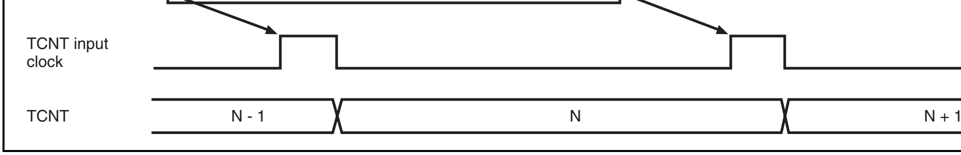


Figure 10.93 Count Timing in Internal Clock Operation (Channels 0 to 4

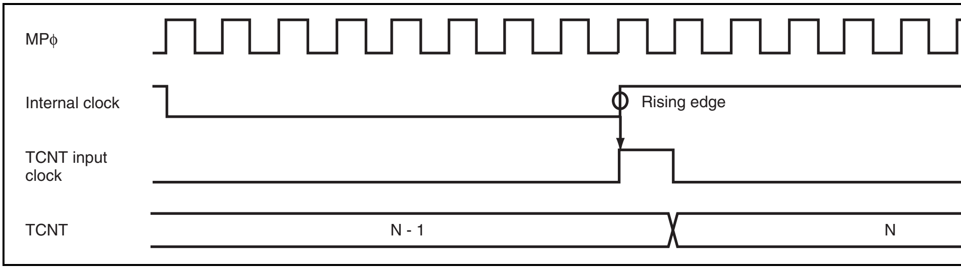


Figure 10.94 Count Timing in Internal Clock Operation (Channel 5)

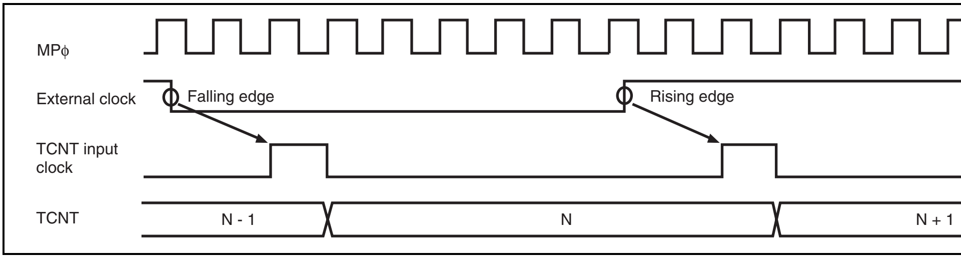


Figure 10.95 Count Timing in External Clock Operation (Channels 0 to 4

which TCNT and TGR match (the point at which the count value matched by TCNT is up
When a compare match signal is generated, the output value set in TIOR is output at the o
compare output pin (TIOC pin). After a match between TCNT and TGR, the compare ma
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 fig
10.98 shows output compare output timing (complementary PWM mode and reset synchr
PWM mode).

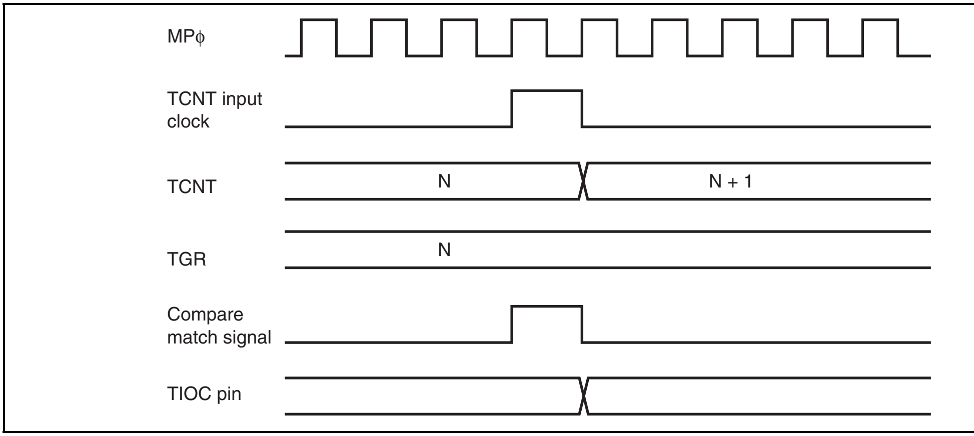


**Figure 10.97 Output Compare Output Timing (Normal Mode/PWM Mode**

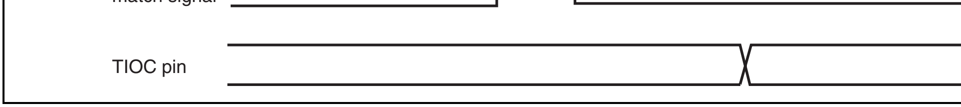


**Figure 10.98 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)**

**Input Capture Signal Timing:** Figure 10.99 shows input capture signal timing.

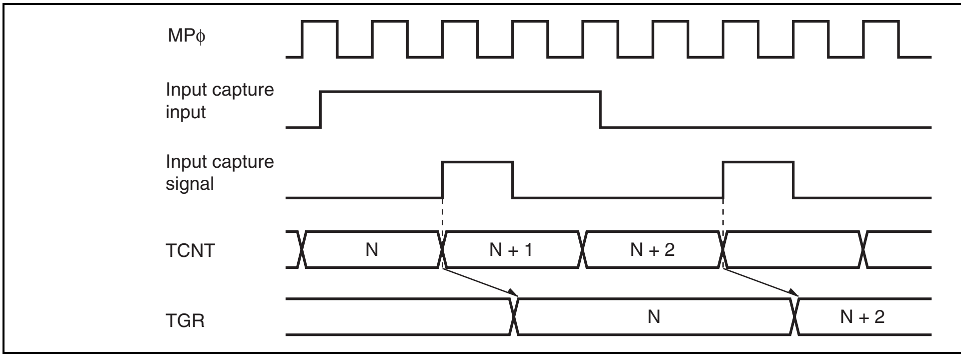


**Figure 10.99 Input Capture Input Signal Timing**

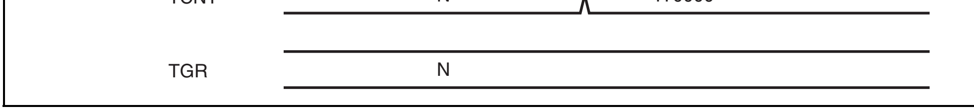


Figure 10.100 Counter Clear Timing (Compare Match) (Channels 0 to 4)

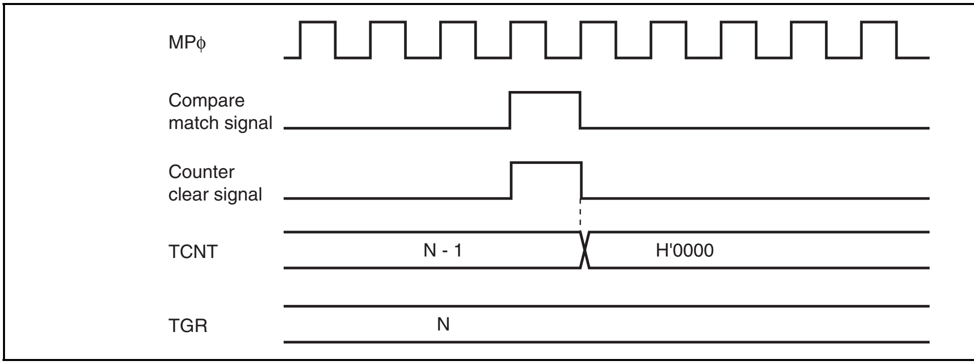


Figure 10.101 Counter Clear Timing (Compare Match) (Channel 5)

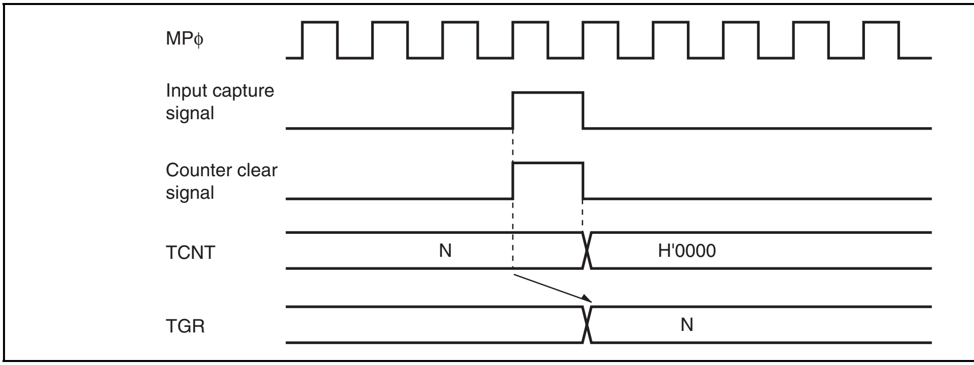


Figure 10.102 Counter Clear Timing (Input Capture) (Channels 0 to 5)

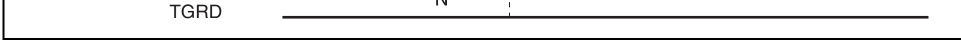


**Figure 10.103   Buffer Operation Timing (Compare Match)**

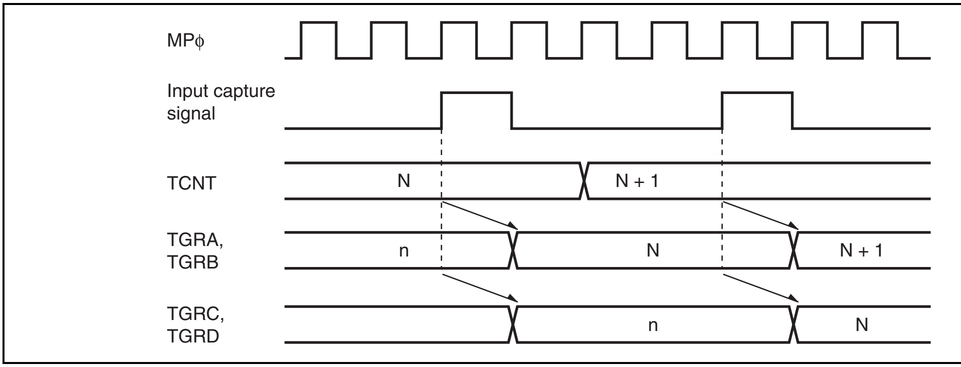


**Figure 10.104   Buffer Operation Timing (Input Capture)**

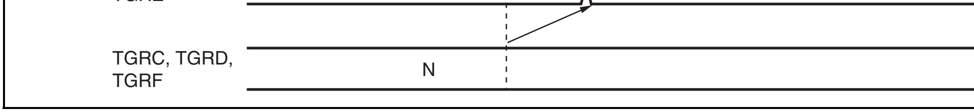


**Figure 10.105 Buffer Transfer Timing (when TCNT Cleared)**

**Buffer Transfer Timing (Complementary PWM Mode)**: Figures 10.106 to 10.108 sho
buffer transfer timing in complementary PWM mode.

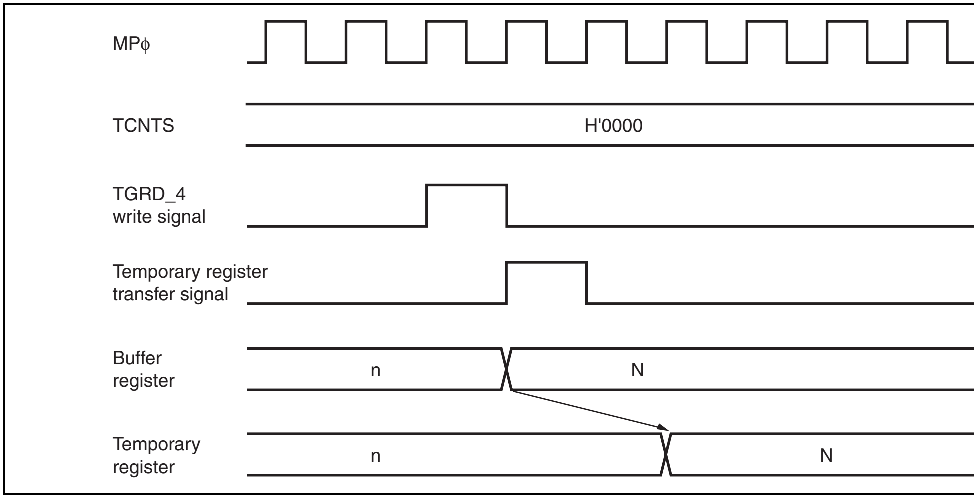


**Figure 10.106 Transfer Timing from Buffer Register to Temporary Register (TCN**

**Figure 10.107 Transfer Timing from Buffer Register to Temporary Regist**
**(TCNTS Operating)**

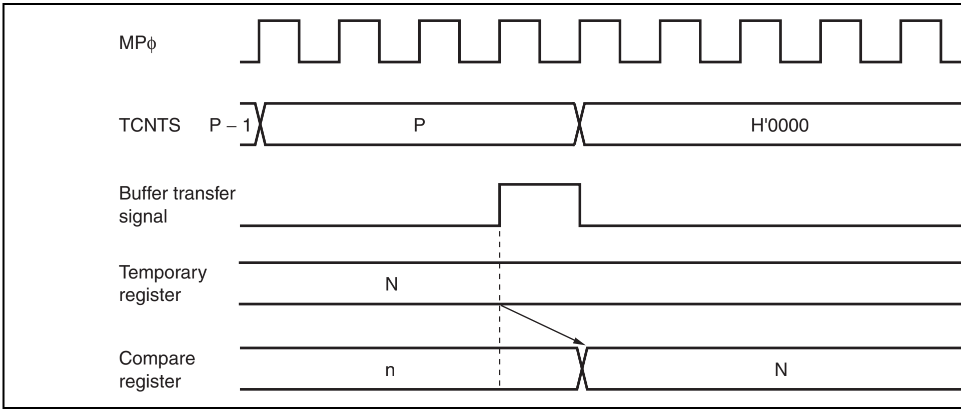


**Figure 10.108 Transfer Timing from Temporary Register to Compare Regi**

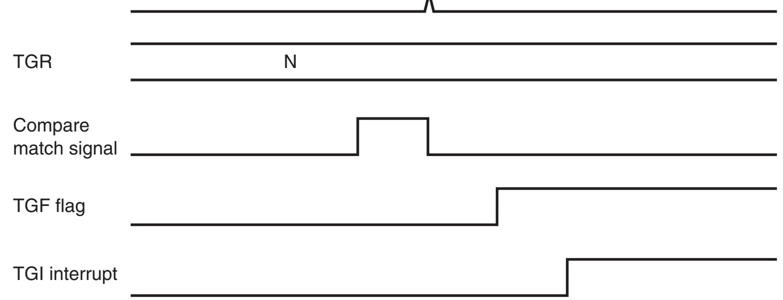


**Figure 10.109 TGI Interrupt Timing (Compare Match) (Channels 0 to 4)**

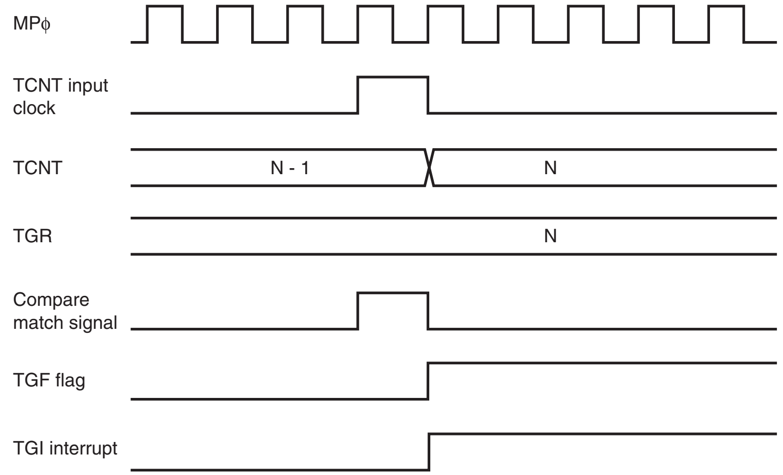


**Figure 10.110 TGI Interrupt Timing (Compare Match) (Channel 5)**

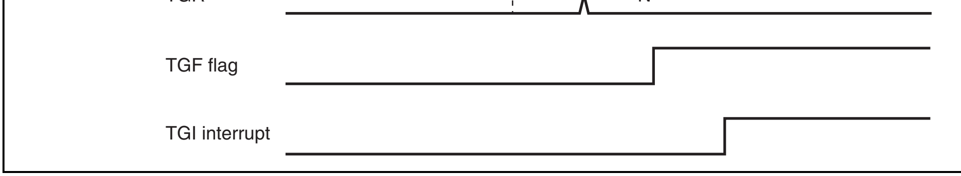


**Figure 10.111   TGI Interrupt Timing (Input Capture) (Channels 0 to 4)**

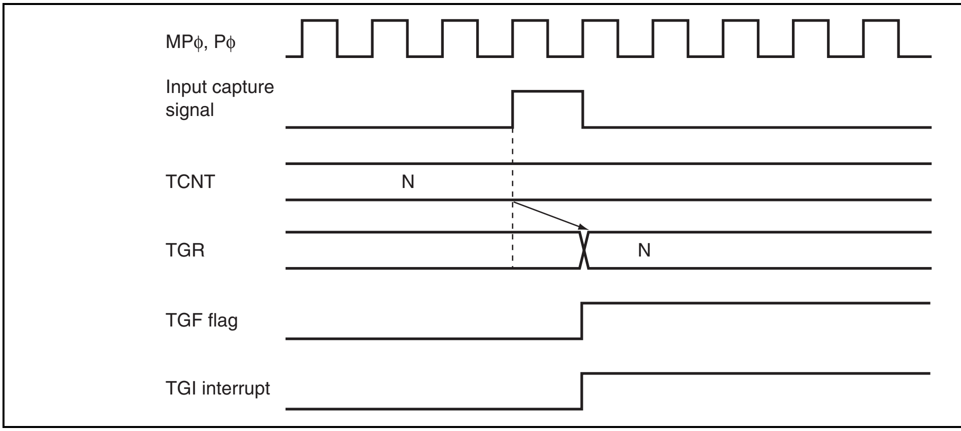


**Figure 10.112   TGI Interrupt Timing (Input Capture) (Channel 5)**

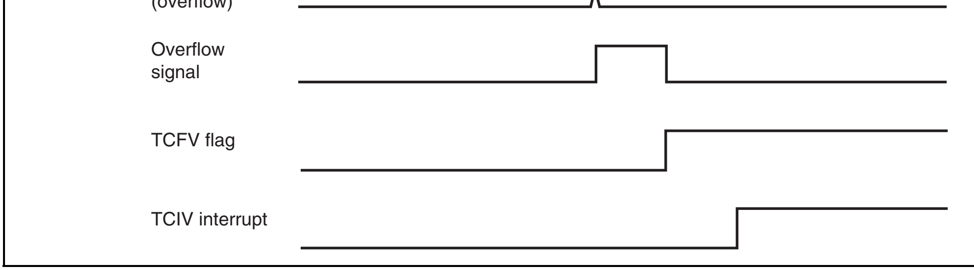


Figure 10.113 TCIV Interrupt Setting Timing

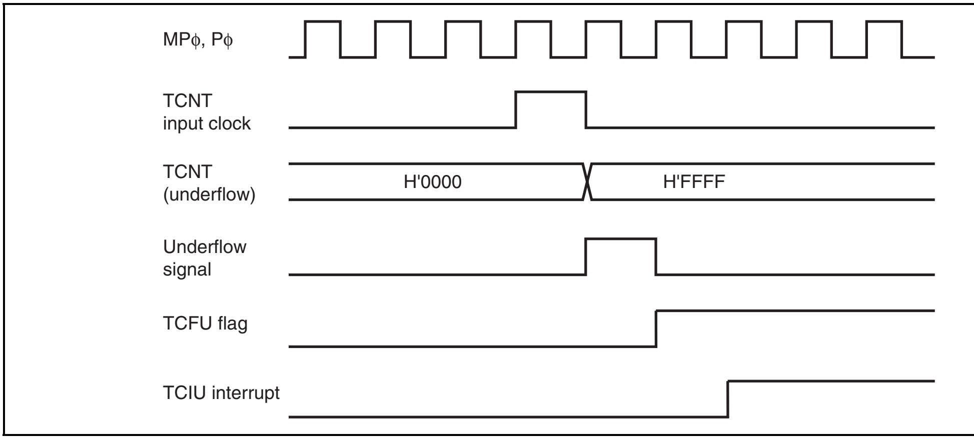


Figure 10.114 TCIU Interrupt Setting Timing

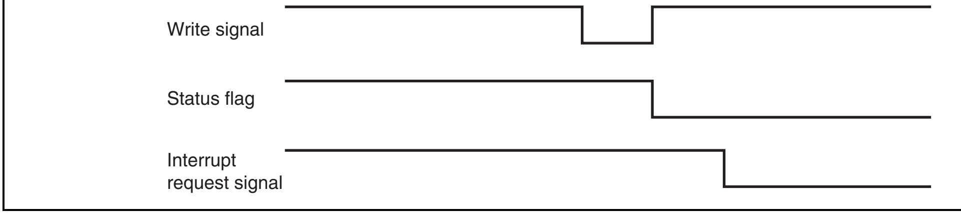


**Figure 10.115 Timing for Status Flag Clearing by CPU (Channels 0 to 4)**

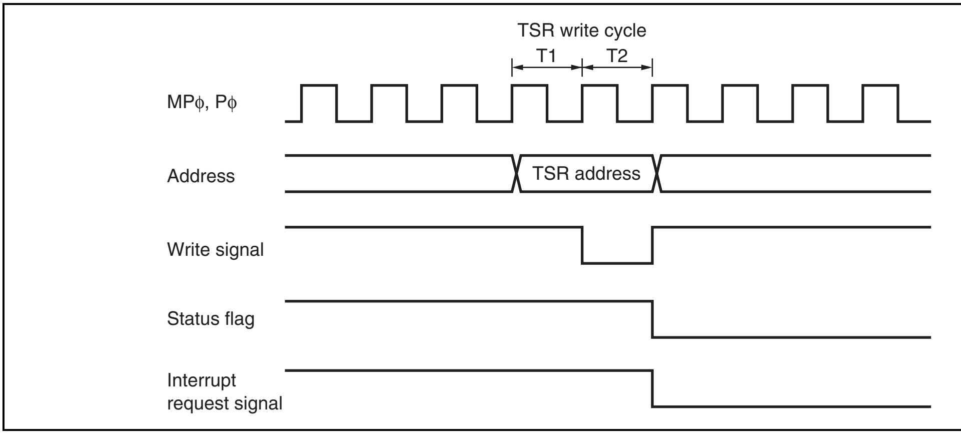


**Figure 10.116 Timing for Status Flag Clearing by CPU (Channel 5)**

Figure 10.117 Timing for Status Flag Clearing by DTC Activation (Channels 0

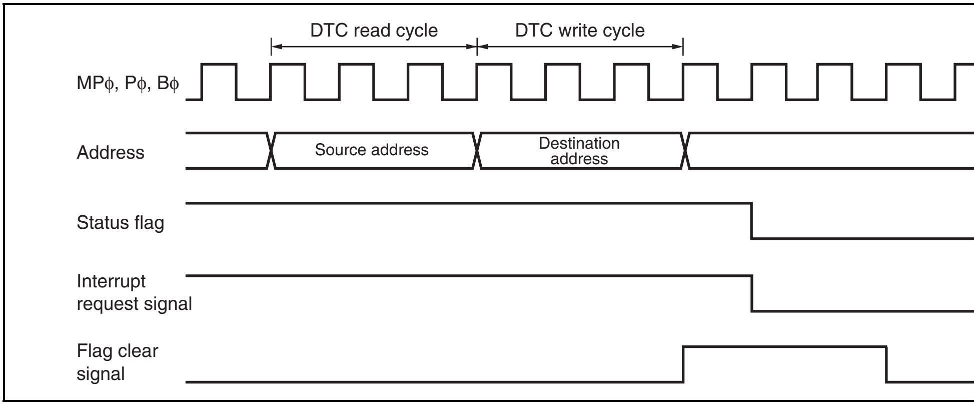
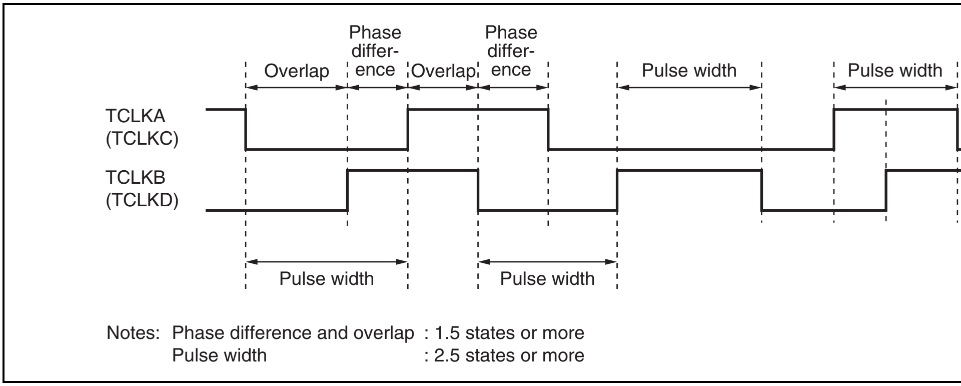


Figure 10.118 Timing for Status Flag Clearing by DTC Activation (Channel

least 2.5 states in the case of both-edge detection. The MTU2 will not operate properly a

pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.119 shows the i 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**

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

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 clear
precedence and the TCNT write is not performed.

Figure 10.120 shows the timing in this case.

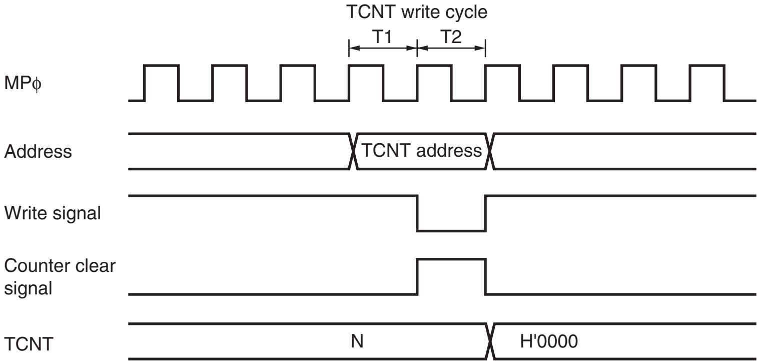


**Figure 10.120 Contention between TCNT Write and Clear Operations**

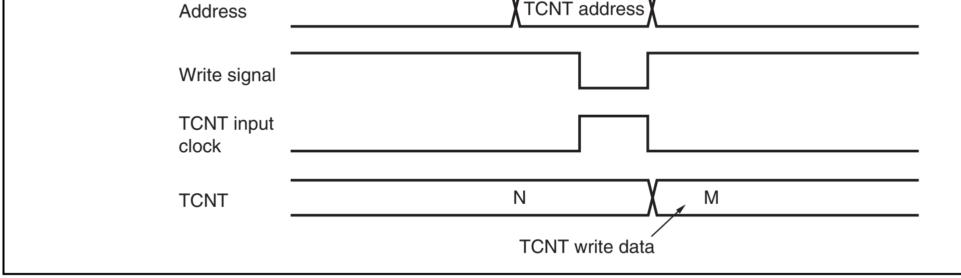


**Figure 10.121   Contention between TCNT Write and Increment Operatio**

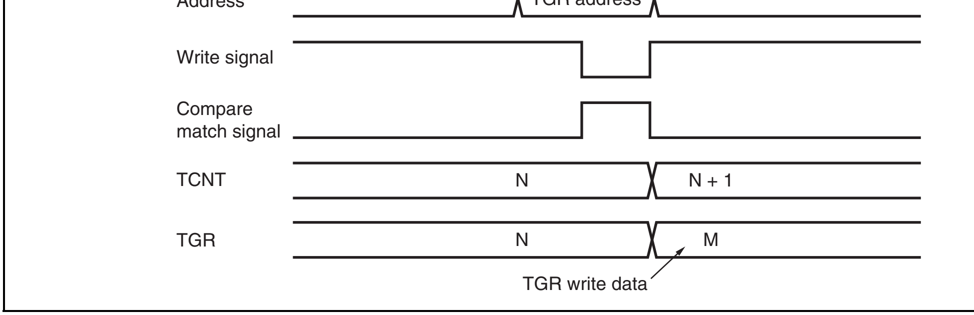


**Figure 10.122 Contention between TGR Write and Compare Match**

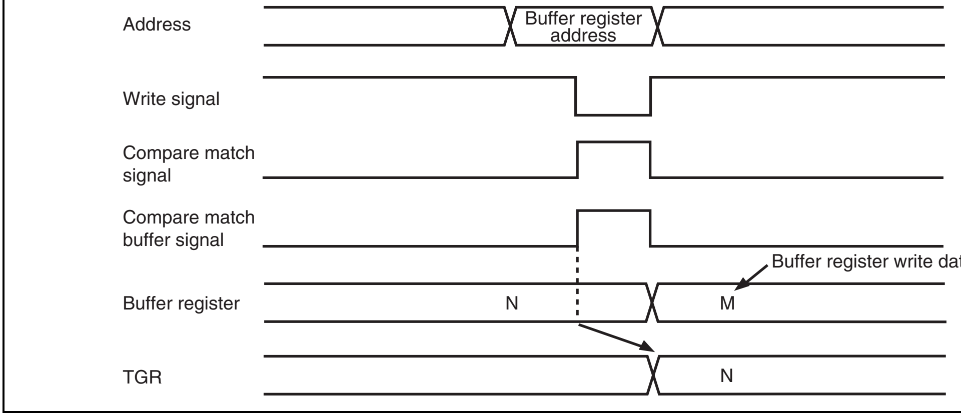


**Figure 10.123 Contention between Buffer Register Write and Compare Ma**

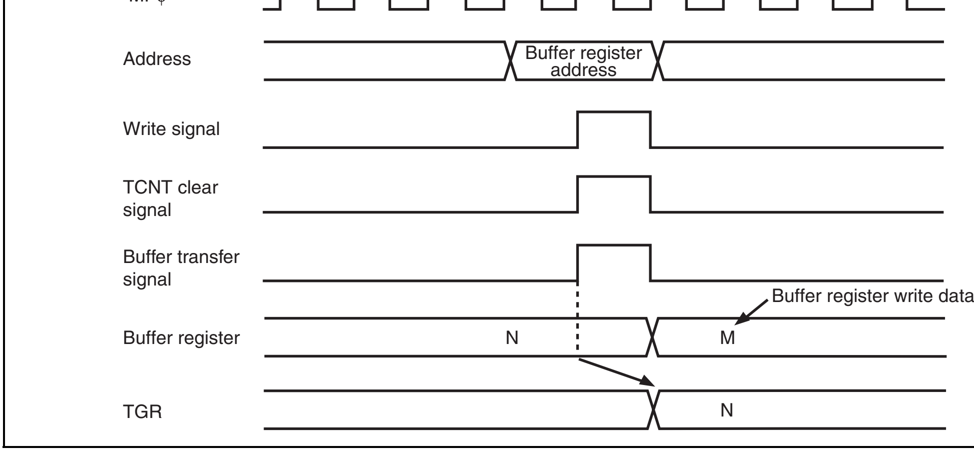


**Figure 10.124 Contention between Buffer Register Write and TCNT Clear**

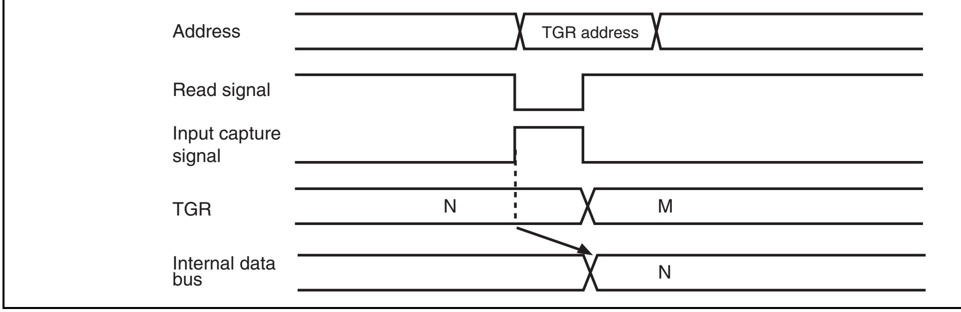


**Figure 10.125 Contention between TGR Read and Input Capture (Channels (**

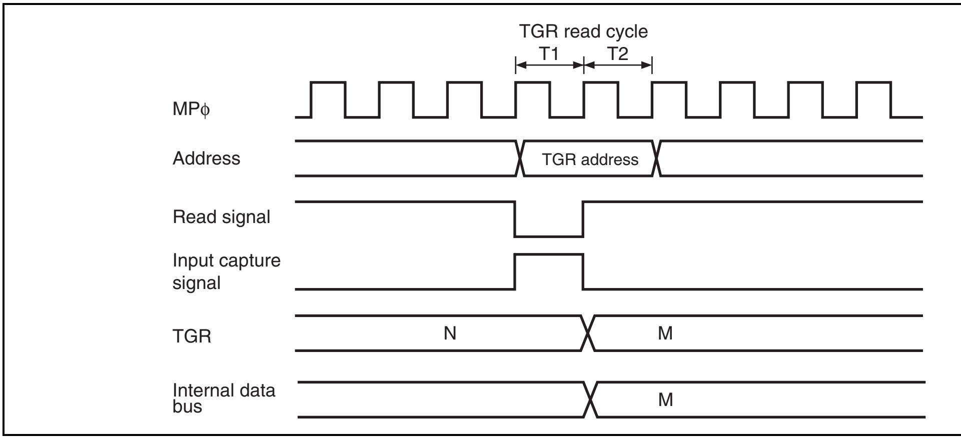


**Figure 10.126 Contention between TGR Read and Input Capture (Channe**

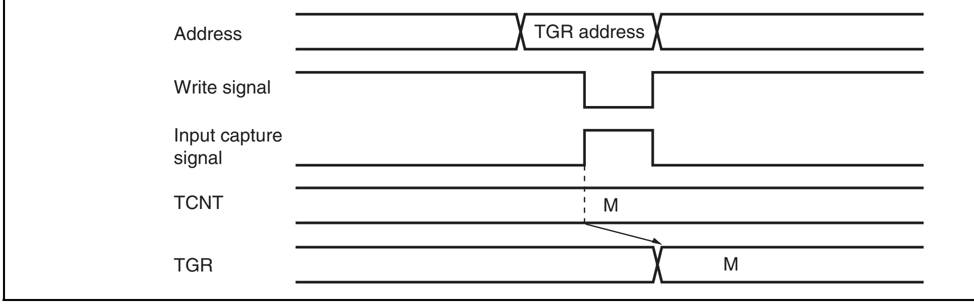


**Figure 10.127 Contention between TGR Write and Input Capture (Channels 0**

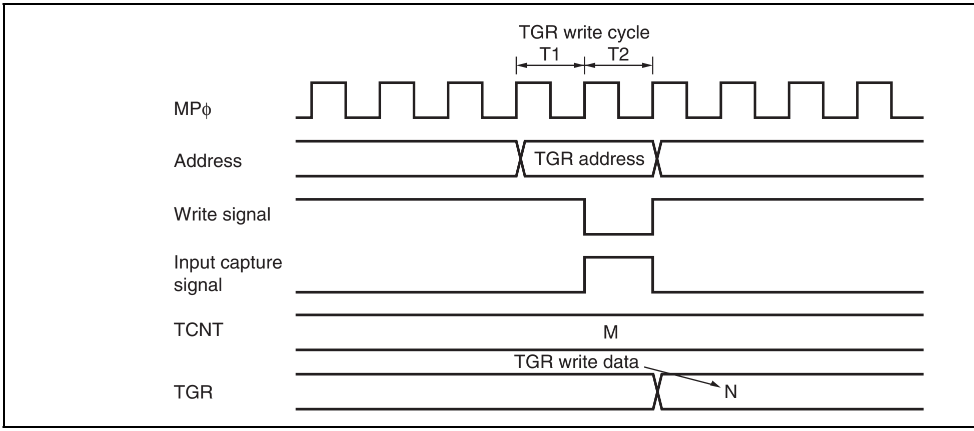


**Figure 10.128 Contention between TGR Write and Input Capture (Channel**

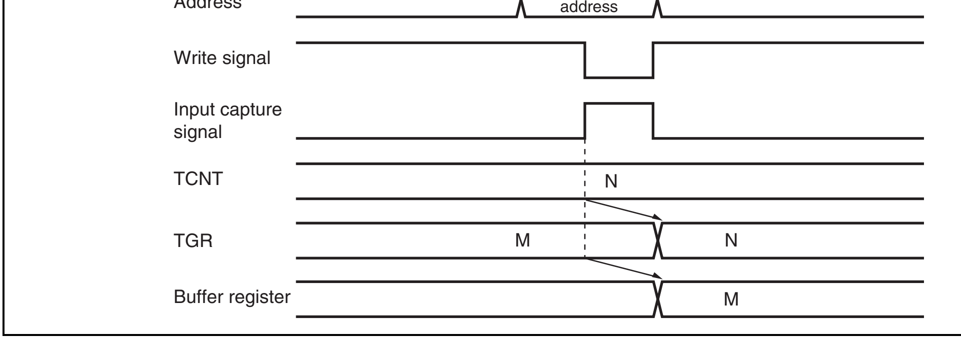


**Figure 10.129 Contention between Buffer Register Write and Input Captu**

### 10.7.12 TCNT_2 Write and Overflow/Underflow Contention in Cascade Connect

With timer counters TCNT_1 and TCNT_2 in a cascade connection, when a contention during TCNT_1 count (during a TCNT_2 overflow/underflow) in the T2 state of the TC write cycle, the write to TCNT_2 is conducted, and the TCNT_1 count signal is disablec 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 ch TGRA_0 to TGRD_0 carry out the input capture operation. In addition, when the compa match/input capture is selected as the input capture source of TGRB_1, TGRB_1 carries capture operation. The timing is shown in figure 10.130.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setti clearing.

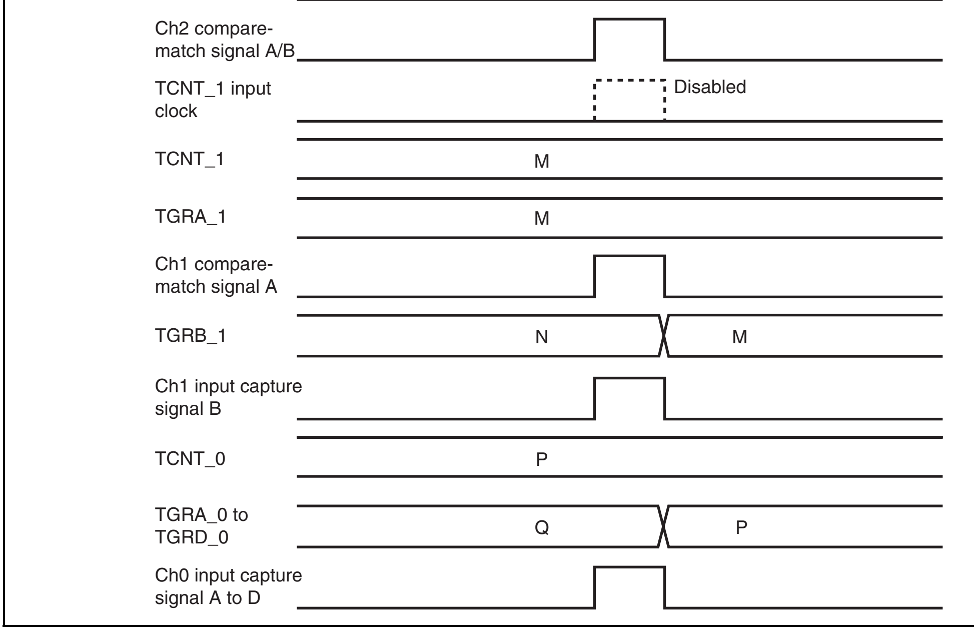

Ch2 compare-match signal A/B
TCNT_1 input clock
Disabled
TCNT_1
M
TGRA_1
M
Ch1 compare-match signal A
TGRB_1
N
M
Ch1 input capture signal B
TCNT_0
P
TGRA_0 to TGRD_0
Q
P
Ch0 input capture signal A to D

**Figure 10.130 TCNT_2 Write and Overflow/Underflow Contention with Casc Connection**

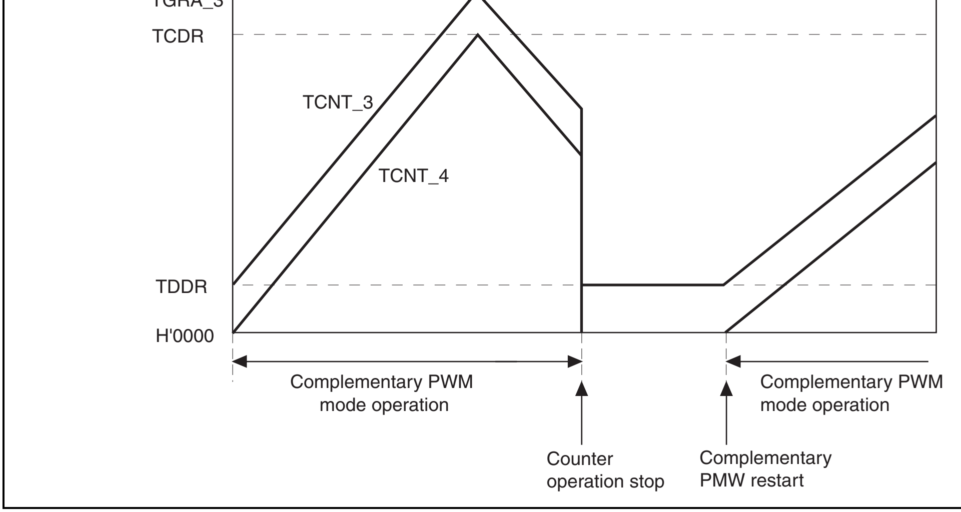


**Figure 10.131 Counter Value during Complementary PWM Mode Stop**

### 10.7.14 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle register (TGRA_3), timer cycle data register (TCDR), and duty setting registers (TGRB_ TGRA_4, and TGRB_4).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance v settings BFA and BFB of TMDR_3. When TMDR_3's BFA bit is set to 1, TGRC_3 fun buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register TGRA_4, and TCBR functions as the TCDR's buffer register.

The TGFC bit and TGFD bit in TSR_3 and TSR_4 are not set when TGRC_3 and TGRD
operating as buffer registers.

Figure 10.132 shows an example of operations for TGR_3, TGR_4, TIOC3, and TIOC4,
TMDR_3's BFA and BFB bits set to 1, and TMDR_4's BFA and BFB bits set to 0.

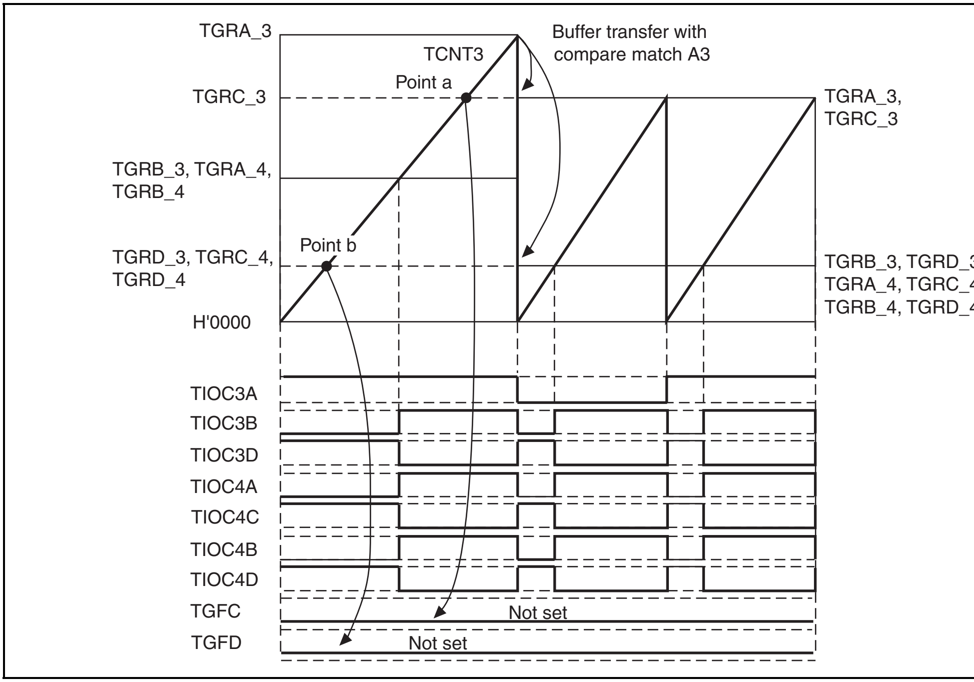


**Figure 10.132 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode**

Figure 10.133 shows a TCFV bit operation example in reset synchronous PWM mode w
value for cycle register TGRA_3 of H'FFFF, when a TGRA_3 compare-match has been
without synchronous setting for the counter clear source.

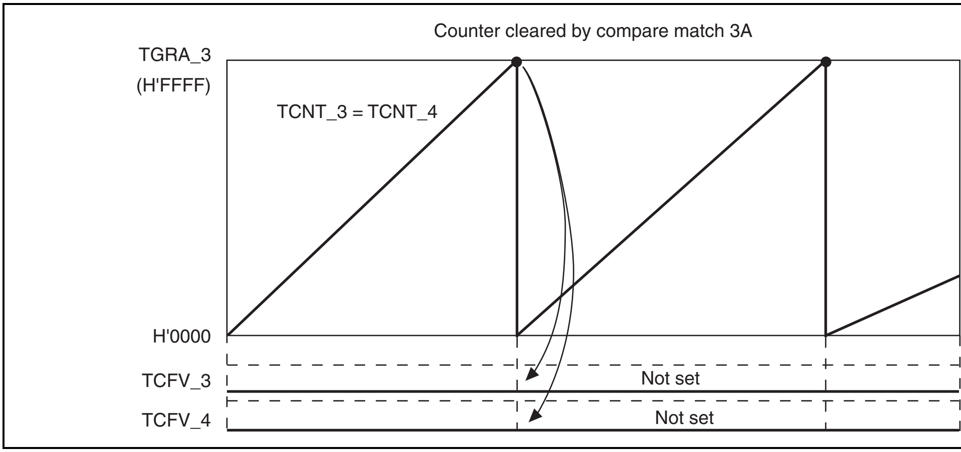


**Figure 10.133 Reset Synchronous PWM Mode Overflow Flag**

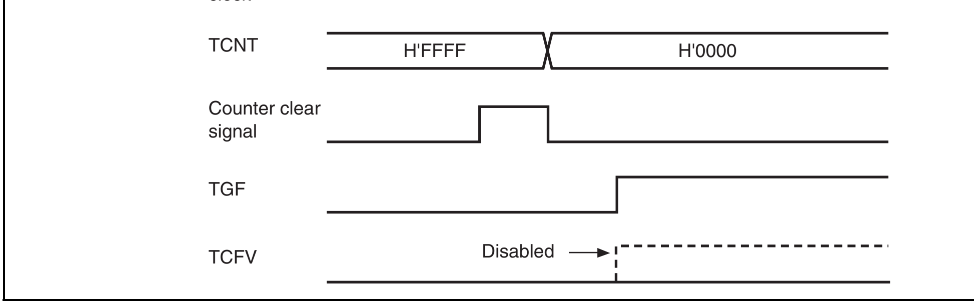


Figure 10.134 Contention between Overflow and Counter Clearing

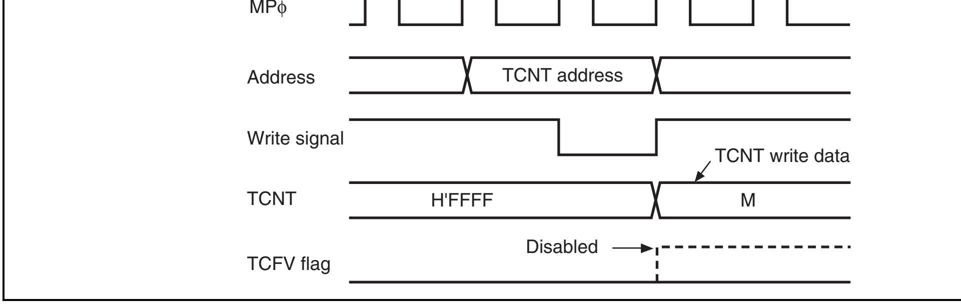


**Figure 10.135 Contention between TCNT Write and Overflow**

### 10.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Rese Synchronized PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to rese synchronized PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3 TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-level state, followed by the transitio synchronized PWM mode and operation in that mode, the initial pin output will not be c

When making a transition from normal operation to reset-synchronized PWM mode, wr registers TIORH_3, TIORL_3, TIORH_4, and TIORL_4 to initialize the output pins to l 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 s normal operation, then initialize the output pins to low level output and set an initial reg of H'00 before making the transition to reset-synchronized PWM mode.

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
connection, the cascade counter value cannot be captured successfully even if input-captu
is simultaneously done to TIOC1A and TIOC2A or to TIOC1B and TIOC2B. This is bec
input timing of TIOC1A and TIOC2A or of TIOC1B and TIOC2B may not be the same v
external input-capture signals to be input into TCNT_1 and TCNT_2 are taken in synchro
with the internal clock. For example, TCNT_1 (the counter for upper 16 bits) does not ca
count-up value by overflow from TCNT_2 (the counter for lower 16 bits) but captures the
value before the count-up. In this case, the values of TCNT_1 = H'FFF1 and TCNT_2 = 
should be transferred to TGRA_1 and TGRA_2 or to TGRB_1 and TGRB_2, but the valu
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 wit
input-capture input as the trigger. This function allows reading of the 32-bit counter such
TCNT_1 and TCNT_2 are captured at the same time. For details, see section, 10.3.8, Tim
Capture Control Register (TICCR).

Condition (1): When synchronous clearing occurs in the PWM output dead time interval initial output suppression interval (10) (figure 10.136).

Condition (2): When synchronous clearing occurs within initial output suppression inter (11) and TGRB_3 ≤ TDDR, TGRA_4 ≤ TDDR, or TGRB_4 ≤ TDDR is (figure 10.137).

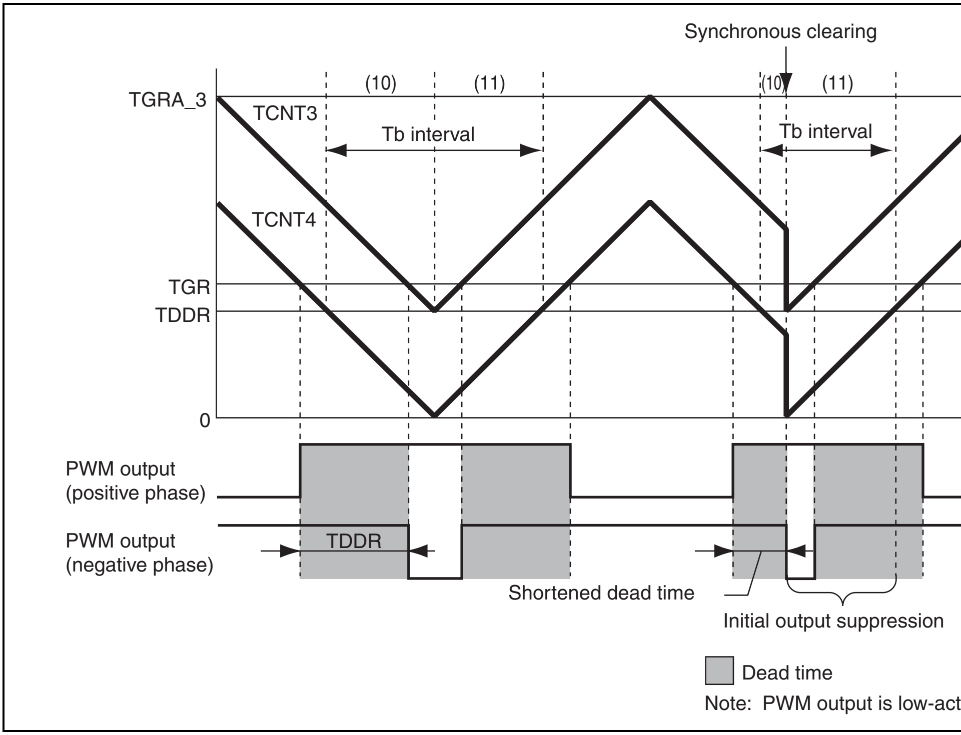
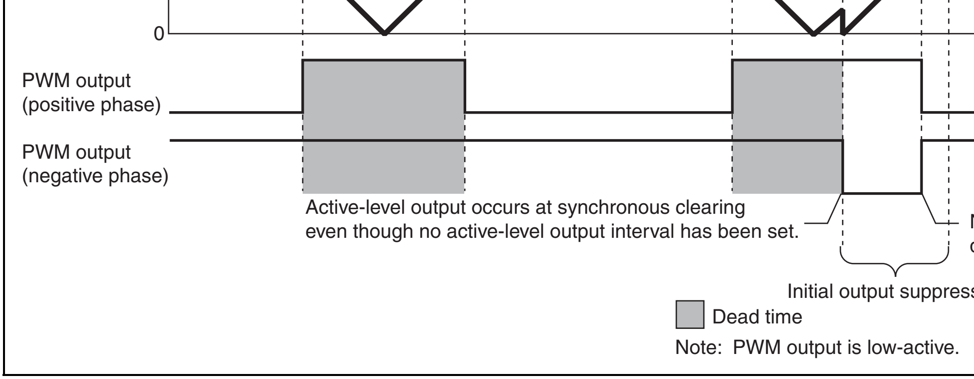


**Figure 10.136 Condition (1) Synchronous Clearing Example**



Active-level output occurs at synchronous clearing even though no active-level output interval has been set.

Initial output suppress

Note: PWM output is low-active.

**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 TGRB_4 to a value twice or more the setting of dead time data register TDDR.

- 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

### 10.8.2 Reset Start Operation

The MTU2 output pins (TIOC*) are initialized low by a reset and in standby mode. Sinc
pin function selection is performed by the pin function controller (PFC), when the PFC i
MTU2 pin states at that point are output to the ports. When MTU2 output is selected by
immediately after a reset, the MTU2 output initial level, low, is output directly at the po
the active level is low, the system will operate at this point, and therefore the PFC settin
be made after initialization of the MTU2 output pins is completed.

Note: Channel number and port notation are substituted for *.

### Table 10.59 Mode Transition Combinations

| | After | | | | | |
|---|---|---|---|---|---|---|
| **Before** | **Normal** | **PWM1** | **PWM2** | **PCM** | **CPWM** | **RP** |
| Normal | (1) | (2) | (3) | (4) | (5) | (6) |
| PWM1 | (7) | (8) | (9) | (10) | (11) | (12 |
| PWM2 | (13) | (14) | (15) | (16) | None | Nor |
| PCM | (17) | (18) | (19) | (20) | None | Nor |
| 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

not initialize the pins. If initialization is required, carry it out in normal mode, then s
PWM mode 2.

- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, se TIOR will not initialize the buffer register pins. If initialization is required, clear buf carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIO initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out ini then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level selected by the timer output control register (TOCR) setting, switch to normal mode perform initialization with TIOR, then restore TIOR to its initial value, and tempora channel 3 and 4 output with the timer output master enable register (TOER). Then o unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, T setting).

Note: Channel number is substituted for * indicated in this article.

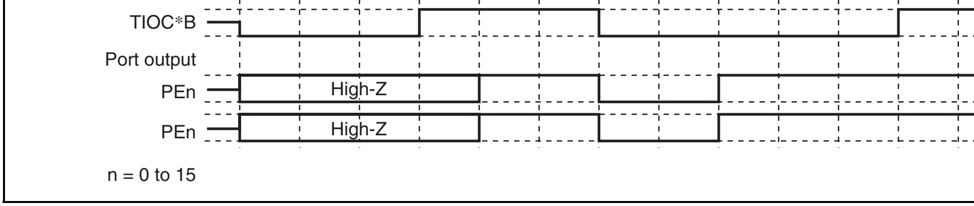


**Figure 10.138 Error Occurrence in Normal Mode, Recovery in Normal Mo**

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 TIO
4. Initialize the pins with TIOR. (The example shows initial high output, with low outp 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.

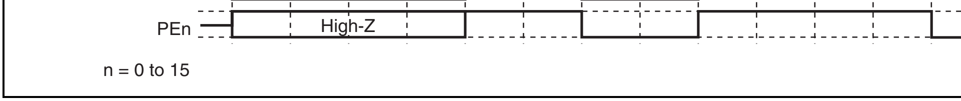


**Figure 10.139 Error Occurrence in Normal Mode, Recovery in PWM Mod**

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. 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.

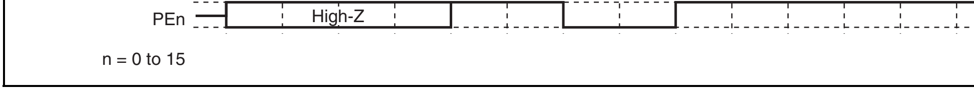


**Figure 10.140 Error Occurrence in Normal Mode, Recovery in PWM Mode**

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 initial 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 n necessary.

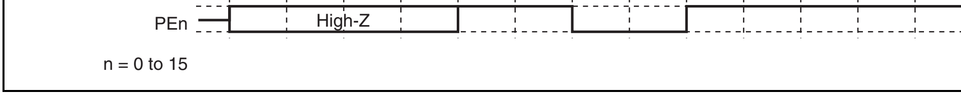


**Figure 10.141 Error Occurrence in Normal Mode, Recovery in Phase Countin**

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 not necessary.

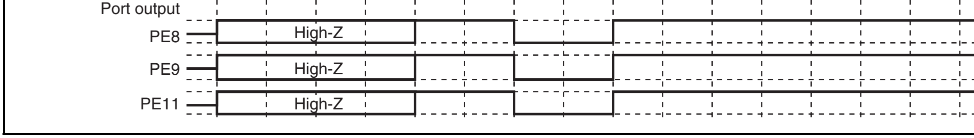


**Figure 10.142 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode**

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 wi 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.

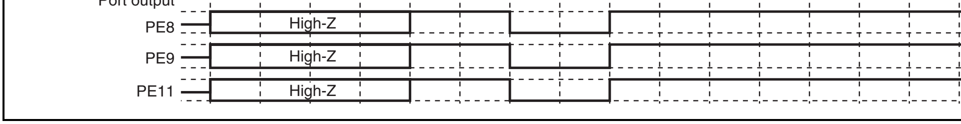


**Figure 10.143 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode**

1 to 13 are the same as in figure 10.138.

14. Select the reset-synchronized PWM output level and cyclic output enabling/disablin 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.

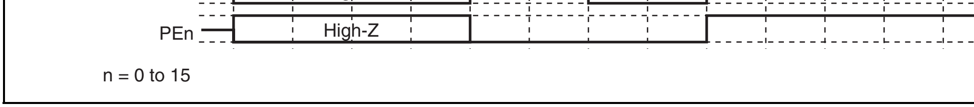


**Figure 10.144 Error Occurrence in PWM Mode 1, Recovery in Normal Mo**

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 TIO
4. Initialize the pins with TIOR. (The example shows initial high output, with low outp 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.

**Figure 10.145   Error Occurrence in PWM Mode 1, Recovery in PWM Mod**

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.

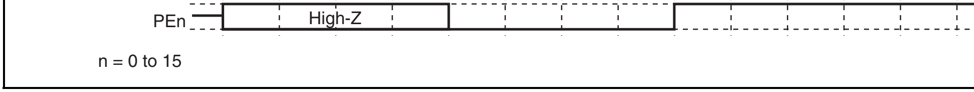


**Figure 10.146 Error Occurrence in PWM Mode 1, Recovery in PWM Mode**

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 initial
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 n necessary.

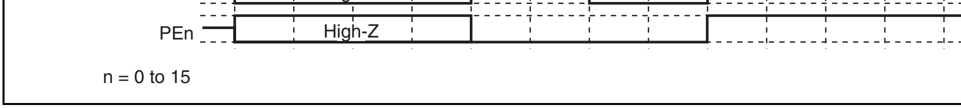


**Figure 10.147 Error Occurrence in PWM Mode 1, Recovery in Phase Countin**

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 not necessary.

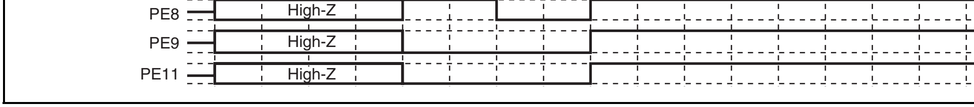


**Figure 10.148 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode**

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 wi 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.

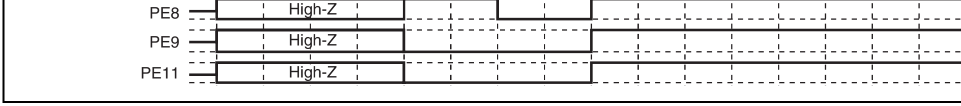


**Figure 10.149 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode**

1 to 14 are the same as in figure 10.148.

15. Select the reset-synchronized PWM output level and cyclic output enabling/disablin 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.

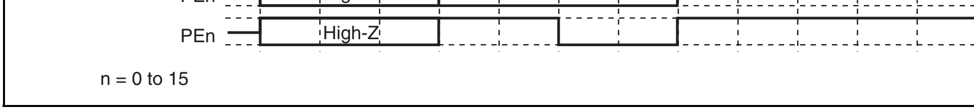


**Figure 10.150 Error Occurrence in PWM Mode 2, Recovery in Normal Mo**

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 outp compare-match occurrence. In PWM mode 2, the cycle register pins are not initialize 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.

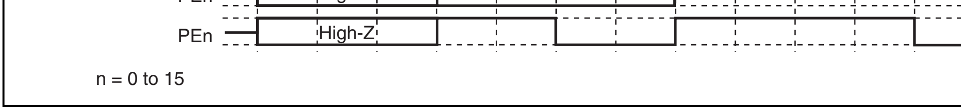


**Figure 10.151   Error Occurrence in PWM Mode 2, Recovery in PWM Mod**

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.

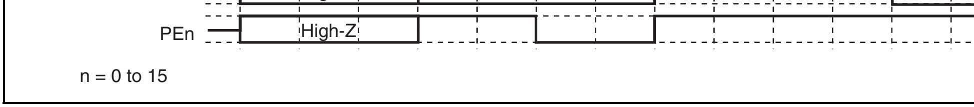


**Figure 10.152 Error Occurrence in PWM Mode 2, Recovery in PWM Mode**

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 initial
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

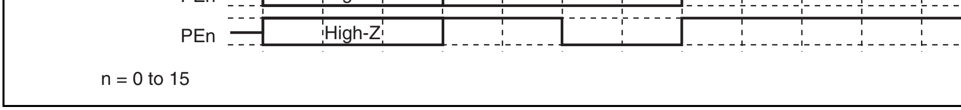


**Figure 10.153 Error Occurrence in PWM Mode 2, Recovery in Phase Countin**

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.

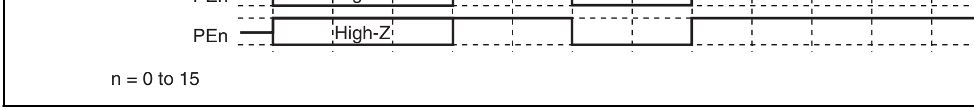


**Figure 10.154 Error Occurrence in Phase Counting Mode, Recovery in Normal**

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 outp 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.

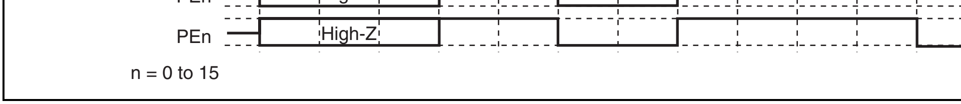


**Figure 10.155 Error Occurrence in Phase Counting Mode, Recovery in PWM**

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.

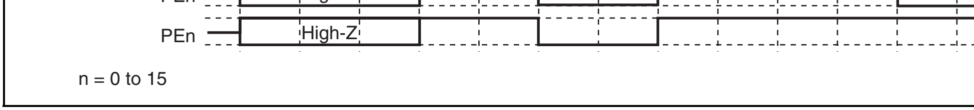


**Figure 10.156 Error Occurrence in Phase Counting Mode, Recovery in PWM M**

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 initial
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

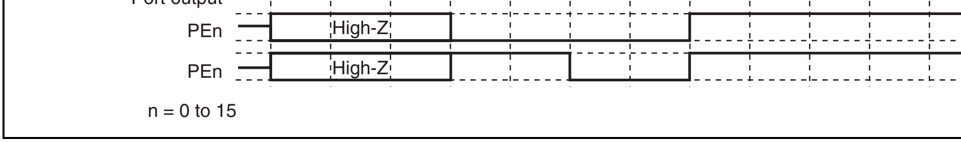


**Figure 10.157 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode**

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.

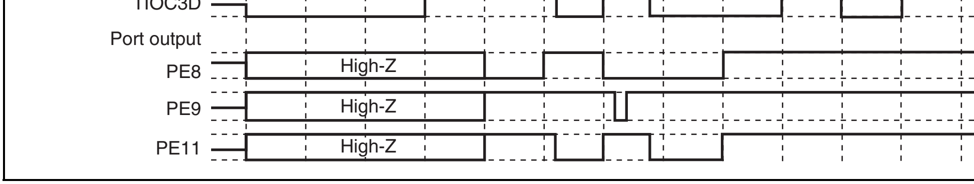


**Figure 10.158 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode**

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 wi 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 complementar 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.

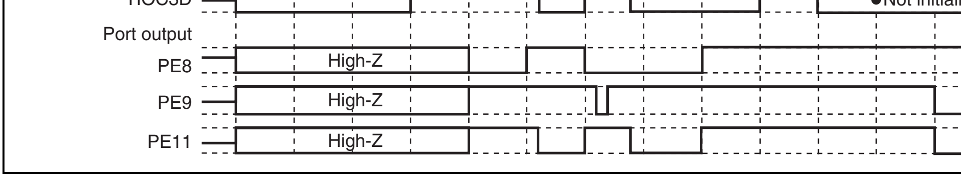


**Figure 10.159 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1**

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.

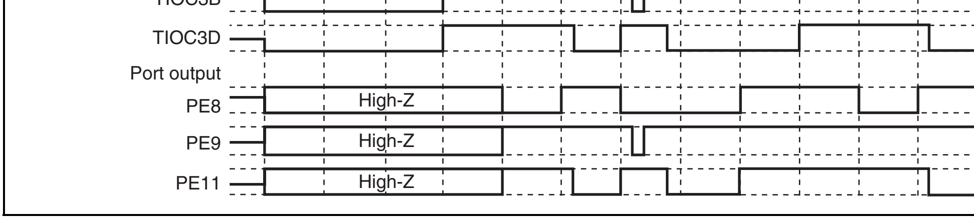


**Figure 10.160 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

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.

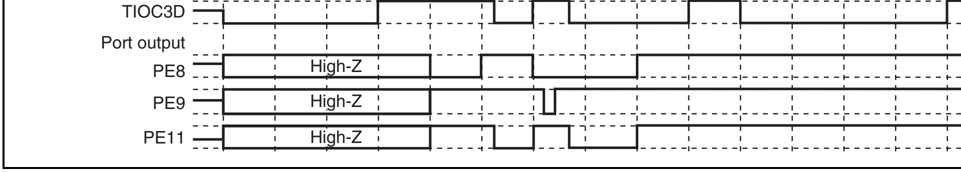


**Figure 10.161 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

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/disab 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.

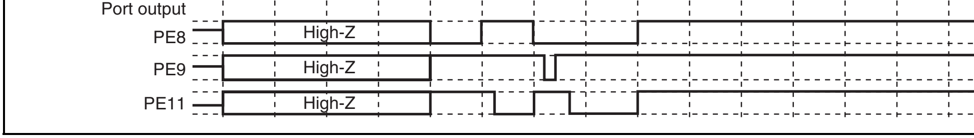


**Figure 10.162 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode**

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/dis 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.

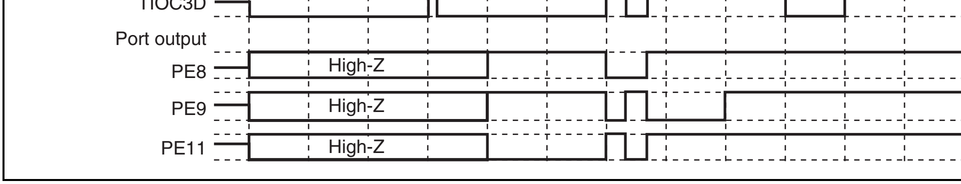


**Figure 10.163 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode**

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/disablin 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-synchro PWM output initial value.)
11. Set normal mode. (MTU2 positive phase output is low, and negative phase output is
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

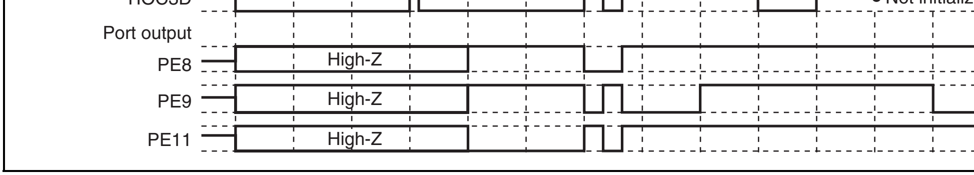


**Figure 10.164 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1**

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
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.

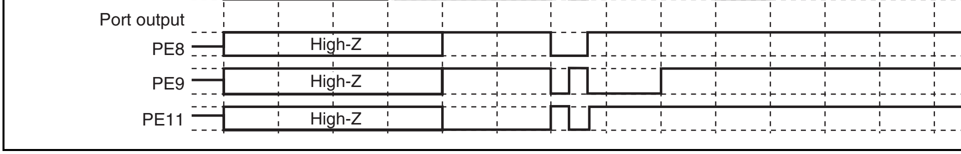


**Figure 10.165 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode**

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 w 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.

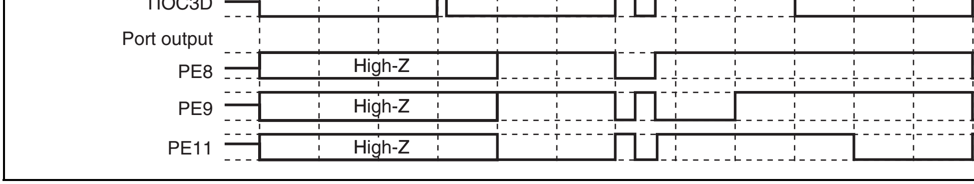


**Figure 10.166 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode**

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.

| | | | | |
|---|---|---|---|---|
| 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 r 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 | | √ | √ | — |

| | | | |
|---|---|---|---|
| | | (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 r input captu<br>• Compare r input captu<br>• Compare r input captu |
| 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 | — |

|  |  |  |  |
|---|---|---|---|
|  | TIOC4DS | I/O | TGRD_4S input capture input/output compare output/PWM |
| 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 |

| | | | | | |
|---|---|---|---|---|---|
| Timer control register_3S | TCR_3S | R/W | H'00 | H'FFFFC600 | 8, 1 |
| 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, 1 |
| 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, 1 |
| 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, 1 |
| 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, 1 |
| 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, 1 |
| 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, |
| 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, |
| 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, |
| 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, |
| Timer general register B_4S | TGRB_4S | R/W | H'FFFF | H'FFFFC61E | 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, |
| 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, |
| 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 |
| 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 |
| Timer A/D converter start request cycle set buffer register B_4S | TADCOBRB_4S | R/W | H'FFFF | H'FFFFC64A | 16 |

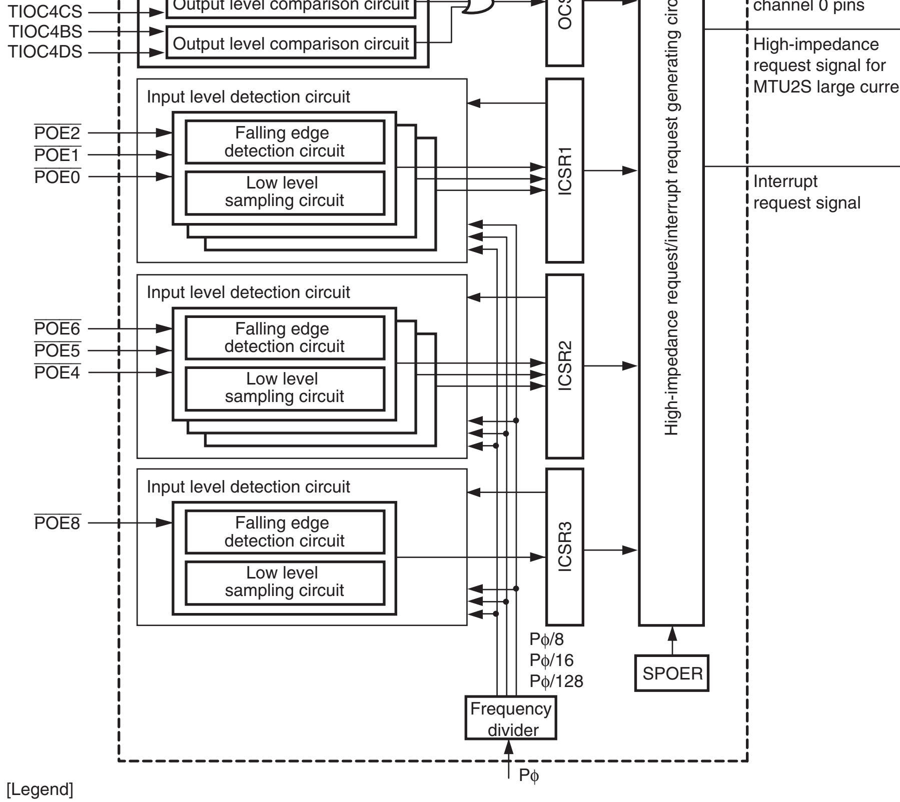
| | | | | | |
|---|---|---|---|---|---|
| Timer counter U_5S | TCNTU_5S | R/W | H'0000 | H'FFFFC880 | 16, |
| Timer general register U_5S | TGRU_5S | R/W | H'FFFF | H'FFFFC882 | 16 |
| Timer control register U_5S | TCRU_5S | R/W | H'00 | H'FFFFC884 | 8 |
| Timer I/O control register U_5S | TIORU_5S | R/W | H'00 | H'FFFFC886 | 8 |
| Timer counter V_5S | TCNTV_5S | R/W | H'0000 | H'FFFFC890 | 16, |
| 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, |
| 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 |

## 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 falli
  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 h
  impedance state on the falling edge or low-level sampling of the $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, PO
  $\overline{\text{POE6}}$, and $\overline{\text{POE8}}$ pins.
- Output levels on the large current pins are compared and if active-level outputs conti
  multiple pins simultaneously for one cycle or more, the large current pins can be pla
  high-impedance state.
- The large current pins and the pins for channel 0 of the MTU2 can be placed in the h
  impedance state by modifying the POE register setting.
- Interrupts can be generated by input-level sampling or output-level comparison resul

The POE has input level detection circuits, output level comparison circuits, and a high-
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-imped
when the oscillator stops or in software standby state. For details, refer to appendix A, P



[Legend]

| | |
|---|---|
| ICSR1: | Input level control/status register 1 |
| ICSR2: | Input level control/status register 2 |
| ICSR3: | Input level control/status register 3 |
| OCSR1: | Output level control/status register 1 |
| OCSR2: | Output level control/status register 2 |
| SPOER: | Software port output enable register |
| POECR1: | Port output enable control register 1 |
| POECR2: | Port output enable control register 2 |

**Figure 12.1 Block Diagram of POE**

| | | | |
|---|---|---|---|
| Port output enable input pin 8 | POE8 | Input | Inputs a request signal to p for channel 0 in MTU2 in th impedance state |

| | | |
|---|---|---|
| | | This active level comparison is done wher MTU2 output function or general output fu selected in the pin function controller. If ar function is selected, the output level is not checked.<br>Pin combinations for output comparison a impedance control can be selected by PO registers. |
| PE16/TIOC3BS and PE17/TIOC3DS | Output | The large current pins for the MTU2S are the high-impedance state when the pins simultaneously output an active level (low when the output level select P (OLSP) bit timer output control register (TOCR) in the is 0 or high level when the bit is 1) for one cycles of the peripheral clock (Pφ).<br>This active level comparison is done wher MTU2S output function or general output is selected in the pin function controller. If function is selected, the output level is not checked.<br>Pin combinations for output comparison a impedance control can be selected by PO registers. |
| PE18/TIOC4AS and PE20/TIOC4CS | | |
| PE19/TIOC4BS and PE21/TIOC4DS | | |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Output level control/status register 1 | OCSR1 | R/W | H'0000 | H'FFFFD002 | 8, |
| Input level control/status register 2 | ICSR2 | R/W | H'0000 | H'FFFFD004 | 8, |
| Output level control/status register 2 | OCSR2 | R/W | H'0000 | H'FFFFD006 | 8, |
| Input level control/status register 3 | ICSR3 | R/W | H'0000 | H'FFFFD008 | 8, |
| 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, |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value sho<br>always be 0. |
| 14 | POE2F | 0 | R/(W)*[1] | POE2 Flag<br>This flag indicates that a high impedance reques<br>been input to the $\overline{\text{POE2}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE2F after reading POE2F<br>(when the falling edge is selected by bits 5 a<br>ICSR1)<br>• By writing 0 to POE2F after reading POE2F<br>a high level input to POE2 is sampled at Pϕ/8<br>or Pϕ/128 clock (when low-level sampling is s<br>by bits 5 and 4 in ICSR1)<br>[Setting condition]<br>• When the input set by ICSR1 bits 5 and 4 oc<br>the $\overline{\text{POE2}}$ pin |

| | | | | |
|---|---|---|---|---|
| | | | | a high level input to POE1 is sampled at P$\phi$<br>or P$\phi$/128 clock (when low-level sampling is<br>by bits 3 and 2 in ICSR1)<br>[Setting condition]<br>• When the input set by ICSR1 bits 3 and 2 o<br>the $\overline{\text{POE1}}$ pin |
| 12 | POE0F | 0 | R/(W)$*^1$ | POE0 Flag<br>This flag indicates that a high impedance reque<br>been input to the $\overline{\text{POE0}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE0F after reading POE0F<br>(when the falling edge is selected by bits 1 a<br>ICSR1)<br>• By writing 0 to POE0F after reading POE0F<br>a high level input to POE0 is sampled at P$\phi$<br>or P$\phi$/128 clock (when low-level sampling is<br>by bits 1 and 0 in ICSR1)<br>[Setting condition]<br>• When the input set by ICSR1 bits 1 and 0 o<br>the $\overline{\text{POE0}}$ pin |
| 11 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu<br>always be 0. |

| | | | | |
|---|---|---|---|---|
| 5, 4 | POE2M[1:0] | 00 | R/W | POE2 mode 1, 0<br>These bits select the input mode of the $\overline{\text{POE2}}$ pi<br>00: Accept request on falling edge of POE2 inpu<br>01: Accept request when POE2 input has been s for 16 P$\phi$/8 clock pulses and all are low level<br>10: Accept request when POE2 input has been s for 16 P$\phi$/16 clock pulses and all are low leve<br>11: Accept request when POE2 input has been s for 16 P$\phi$/128 clock pulses and all are low lev |
| 3, 2 | POE1M[1:0] | 00 | R/W*2 | POE1 mode 1, 0<br>These bits select the input mode of the $\overline{\text{POE1}}$ pi<br>00: Accept request on falling edge of POE1 inpu<br>01: Accept request when POE1 input has been s for 16 P$\phi$/8 clock pulses and all are low level<br>10: Accept request when POE1 input has been s for 16 P$\phi$/16 clock pulses and all are low leve<br>11: Accept request when POE1 input has been s for 16 P$\phi$/128 clock pulses and all are low lev |
| 1, 0 | POE0M[1:0] | 00 | R/W*2 | POE0 mode 1, 0<br>These bits select the input mode of the $\overline{\text{POE0}}$ pi<br>00: Accept request on falling edge of POE0 inpu<br>01: Accept request when POE0 input has been s for 16 P$\phi$/8 clock pulses and all are low level<br>10: Accept request when POE0 input has been s for 16 P$\phi$/16 clock pulses and all are low leve<br>11: Accept request when POE0 input has been s for 16 P$\phi$/128 clock pulses and all are low lev |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed w

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 pai<br>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 =<br>[Setting condition]<br>• When any one of the three pairs of 2-phase has simultaneously become an active level |
| 14 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu<br>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 th<br>impedance state when the OSF1 bit in OCSR1<br>0: Does not place the pins in the high-impedan<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 v<br>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 valu<br>always be 0. |

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

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 shou<br>always be 0. |
| 14 | POE6F | 0 | R/(W)*[1] | POE6 Flag<br>This flag indicates that a high impedance reques<br>been input to the $\overline{\text{POE6}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE6F after reading POE6F<br>(when the falling edge is selected by bits 5 a<br>ICSR2)<br>• By writing 0 to POE6F after reading POE6F<br>a high level input to POE6 is sampled at $P\phi$/8<br>or $P\phi$/128 clock (when low-level sampling is<br>by bits 5 and 4 in ICSR2)<br>[Setting condition]<br>• When the input condition set by bits 5 and 4<br>occurs at the $\overline{\text{POE6}}$ pin |

| | | | | |
|---|---|---|---|---|
| | | | | a high level input to POE5 is sampled at Pϕ<br>or Pϕ/128 clock (when low-level sampling is<br>by bits 3 and 2 in ICSR2)<br>[Setting condition]<br>• When the input condition set by bits 3 and 2<br>occurs at the $\overline{\text{POE5}}$ pin |
| 12 | POE4F | 0 | R/(W)*[1] | POE4 Flag<br>This flag indicates that a high impedance reque<br>been input to the $\overline{\text{POE4}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE4F after reading POE4F<br>(when the falling edge is selected by bits 1 a<br>ICSR2)<br>• By writing 0 to POE4F after reading POE4F<br>a high level input to POE4 is sampled at Pϕ<br>or Pϕ/128 clock (when low-level sampling is<br>by bits 1 and 0 in ICSR2)<br>[Setting condition]<br>• When the input condition set by bits 1 and 0<br>occurs at the $\overline{\text{POE4}}$ pin |
| 11 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu<br>always be 0. |
| 8 | PIE2 | 0 | R/W | Port Interrupt Enable 2<br>This bit enables/disables interrupt requests whe<br>of the POE4F to POE6F bits of the ICSR2 is se<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |

| | | | | |
|---|---|---|---|---|
| | | | | 10: Accept request when POE6 input has been s<br>for 16 P$\phi$/16 clock pulses and all are at a low<br>11: Accept request when POE6 input has been s<br>for 16 P$\phi$/128 clock pulses and all are at a lo |
| 3, 2 | POE5M[1:0] | 00 | R/W*[2] | POE5 mode 1 and 0<br>These bits select the input mode of the $\overline{\text{POE5}}$ pi<br>00: Accept request on falling edge of POE5 inpu<br>01: Accept request when POE5 input has been s<br>for 16 P$\phi$/8 clock pulses and all are at a low<br>10: Accept request when POE5 input has been s<br>for 16 P$\phi$/16 clock pulses and all are at a low<br>11: Accept request when POE5 input has been s<br>for 16 P$\phi$/128 clock pulses and all are at a lo |
| 1, 0 | POE4M[1:0] | 00 | R/W*[2] | POE4 mode 1 and 0<br>These bits select the input mode of the $\overline{\text{POE4}}$ pi<br>00: Accept request on falling edge of POE4 inpu<br>01: Accept request when POE4 input has been s<br>for 16 P$\phi$/8 clock pulses and all are at a low l<br>10: Accept request when POE4 input has been s<br>for 16 P$\phi$/16 clock pulses and all are at a low<br>11: Accept request when POE4 input has been s<br>for 16 P$\phi$/128 clock pulses and all are at a lo |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed w

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 pai 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 =<br>[Setting condition]<br>• When any one of the three pairs of 2-phase has simultaneously become an active level |
| 14 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu 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 th impedance state when the OSF2 bit in OCSR2<br>0: Does not place the pins in the high-impedanc<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 w 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 |
|---|---|---|---|---|
| 15 to 13 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write val always be 0. |
| 12 | POE8F | 0 | R/(W)*[1] | POE8 Flag<br>This flag indicates that a high impedance requ been input to the $\overline{\text{POE8}}$ pin.<br>[Clearing conditions]<br>• By writing 0 to POE8F after reading POE8 (when the falling edge is selected by bits 1 ICSR3)<br>• By writing 0 to POE8F after reading POE8 a high level input to POE8 is sampled at P or Pϕ/128 clock (when low-level sampling i by bits 1 and 0 in ICSR3)<br>[Setting condition]<br>• When the input condition set by bits 1 and ICSR3 occurs at the $\overline{\text{POE8}}$ pin |
| 11, 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write val always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | 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 valu<br>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 $\overline{\text{POE8}}$ p<br>00: Accept request on falling edge of POE8 inp<br>01: Accept request when POE8 input has been for 16 $P\phi$/8 clock pulses and all are low leve<br>10: Accept request when POE8 input has been for 16 $P\phi$/16 clock pulses and all are low lev<br>11: Accept request when POE8 input has been for 16 $P\phi$/128 clock pulses and all are low le |

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed v

2. Can be modified only once after a power-on reset.

| | | | | |
|---|---|---|---|---|
| | | | | These bits are always read as 0. The write v should always be 0. |
| 2 | MTU2SHIZ | 0 | R/W | MTU2S Output High-Impedance<br>This bit specifies whether to place the large pins for the MTU2S in the high-impedance s<br>0: Does not place the pins in the high-imped 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 st<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 f channel 0 in the MTU2 in the high-impedan<br>0: Does not place the pins in the high-imped state<br>[Clearing conditions]<br>• Power-on reset<br>• By writing 0 to MTU2CH0HIZ after readi MTU2CH0HIZ = 1<br>1: Places the pins in the high-impedance st<br>[Setting condition]<br>• By writing 1 to MTU2CH0HIZ |

MTU2CH34HIZ = 1

1: Places the pins in the high-impedance stat

[Setting condition]

- By writing 1 to MTU2CH34HIZ

### 12.3.7 Port Output Enable Control Register 1 (POECR1)

POECR1 is an 8-bit readable/writable register that controls high-impedance state of the p

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
|  | - | - | - | - | MTU2<br>PE3ZE | MTU2<br>PE2ZE | MTU2<br>PE1ZE | MTU2<br>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 va should always be 0. |
| 3 | MTU2PE3ZE | 0 | R/W* | MTU2 PE3 High-Impedance Enable<br>This bit specifies whether to place the PE3/T pin for channel 0 in the MTU2 in the high-imp state when either POE8F or MTU2CH0HIZ b to 1.<br>0: Does not place the pin in the high-impedar<br>1: Places the pin in the high-impedance state |

| | | | | |
|---|---|---|---|---|
| | | | | pin for channel 0 in the MTU2 in the high-im<br>state when either POE8F or MTU2CH0HIZ<br>to 1.<br>0: Does not place the pin in the high-impeda<br>1: Places the pin in the high-impedance stat |
| 0 | MTU2PE0ZE | 0 | R/W* | MTU2 PE0 High-Impedance Enable<br>This bit specifies whether to place the PE0/<br>pin for channel 0 in the MTU2 in the high-im<br>state when either POE8F or MTU2CH0HIZ<br>to 1.<br>0: Does not place the pin in the high-impeda<br>1: Places the pin in the high-impedance stat |

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

| Bit | Bit Name | Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s always be 0. |
| 14 | MTU2P1CZE | 1 | R/W* | MTU2 Port 1 Output Comparison/High-Imped Enable<br>This bit specifies whether to compare output the large current pins for the MTU2, PE9/TIC PE11/TIOC3D, and to place them in the high impedance state when the OSF1 bit is set to the OCE1 bit is 1 or when any one of the PO POE1F, POE2F, and MTU2CH34HIZ bits is s<br>0: Does not compare output levels or place th the high-impedance state<br>1: Compares output levels and places the pin high-impedance state |
| 13 | MTU2P2CZE | 1 | R/W* | MTU2 Port 2 Output Comparison/High-Imped Enable<br>This bit specifies whether to compare output the large current pins for the MTU2, PE12/TI and PE14/TIOC4C, and to place them in the impedance state when the OSF1 bit is set to the OCE1 bit is 1 or when any one of the PO POE1F, POE2F, and MTU2CH34HIZ bits is s<br>0: Does not compare output levels or place th the high-impedance state<br>1: Compares output levels and places the pin high-impedance state |

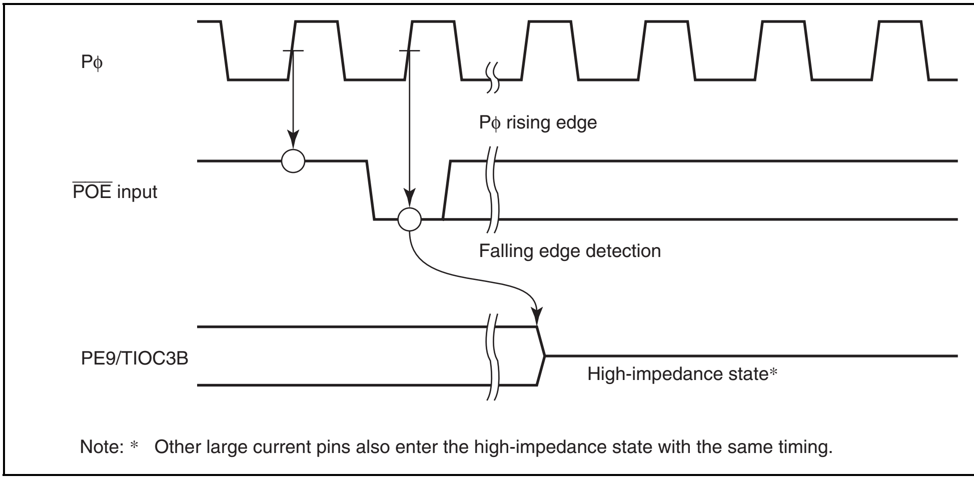
| | | | | |
|---|---|---|---|---|
| | | | | 1: Compares output levels and places the p<br>high-impedance state |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value<br>always be 0. |
| 10 | MTU2SP1CZE | 1 | R/W* | MTU2S Port 1 Output Comparison/High-Im<br>Enable<br>This bit specifies whether to compare outpu the large current pins for the MTU2S, PE16 and PE17/TIOC3DS, and to place them in t impedance state when the OSF2 bit is set t the OCE2 bit is 1 or when any one of the P POE5F, POE6F, and MTU2SHIZ bits is set<br>0: Does not compare output levels or place the high-impedance state<br>1: Compares output levels and places the p high-impedance state |
| 9 | MTU2SP2CZE | 1 | R/W* | MTU2S Port 2 Output Comparison/High-Im<br>Enable<br>This bit specifies whether to compare outpu the large current pins for the MTU2S, PE18 and PE20/TIOC4CS, and to place them in t impedance state when the OSF2 bit is set t the OCE2 bit is 1 or when any one of the P POE5F, POE6F, and MTU2SHIZ bits is set<br>0: Does not compare output levels or place the high-impedance state<br>1: Compares output levels and places the p high-impedance state |

| | | | | |
|---|---|---|---|---|
| | | | | 1: Compares output levels and places the pin<br>high-impedance state |
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va<br>should always be 0. |

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

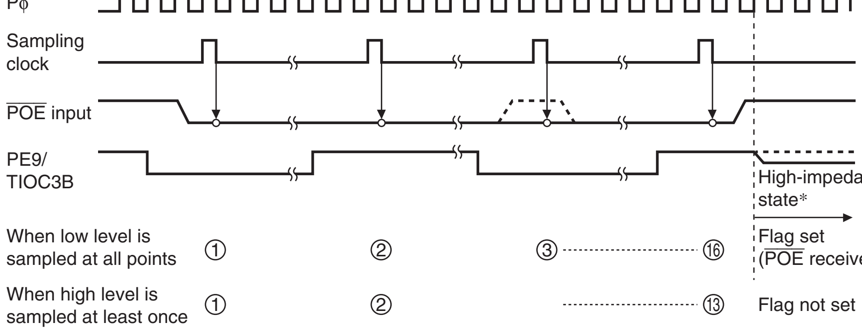
| | | |
|---|---|---|
| the large current pins for the MTU2 (PE12/TIOC4A and PE14/TIOC4C) | Input level detection, output level comparison, or SPOER setting | MTU2P2CZE • ((POE2F + POE1F + POE0F) + (OSF1 • OCE1) + (MTU2CH34 |
| the large current pins for the MTU2 (PE13/TIOC4B and PE15/TIOC4D) | Input level detection, output level comparison, or SPOER setting | MTU2P3CZE • ((POE2F + POE1F + POE0F) + 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) + (MTU2C |
| MTU2 channel 0 pin (PE1/TIOC0B) | Input level detection or SPOER setting | MTU2PE1ZE ((POE8F • POE8E) + (MTU2C |
| MTU2 channel 0 pin (PE2/TIOC0C) | Input level detection or SPOER setting | MTU2PE2ZE ((POE8F • POE8E) + (MTU2C |
| MTU2 channel 0 pin (PE3/TIOC0D) | Input level detection or SPOER setting | MTU2PE3ZE ((POE8F • POE8E) + (MTU2C |

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 to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, and $\overline{\text{POE8}}$ pins until the respective pins enter high-impedance s



Note: * Other large current pins also enter the high-impedance state with the same timing.

**Figure 12.2 Falling Edge Detection**



Note: * Other large current pins also enter the high-impedance state with the same timing.

**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 combinatio TIOC3B and TIOC3D. The operation is the same for the other pin combinations.

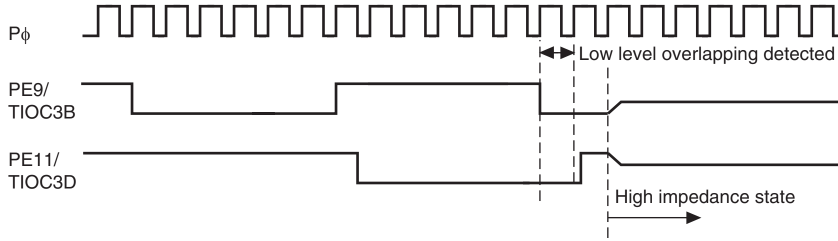


**Figure 12.4 Output-Level Compare Operation**

in bit 15 (OCF1 and OCF2) in OCSR1 and OCSR2. However, note that just writing 0 to a ignored (the flag is not cleared); flags can be cleared only after an inactive level is output large current pins. Inactive-level outputs can be obtained by setting the MTU2 and MTU2 internal registers.

| | | | |
|---|---|---|---|
| OEI2 | Output enable interrupt 2 | POE8F | PIE3 • POE8F |
| OEI3 | Output enable interrupt 3 | POE4F, POE5F, POE6F, and OSF2 | PIE2 • (POE4F + PC<br>POE6F) + OIE2 • OS |

## 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 p 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 a impedance by the port output enable (POE), the pin is placed in the output state for one 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 h 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 impedanc handling with the timer output selected when a power-on reset is issued from the WDT.

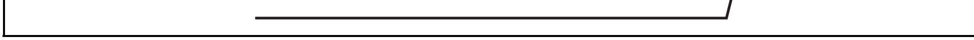


**Figure 12.5   Pin State when a Power-On Reset is Issued from the Watchdog Ti**

## 13.1 Features

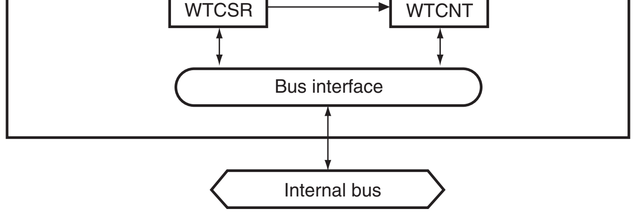
- Can be used to ensure the clock settling time: Use the WDT to revoke software stanc
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode: Internal resets occur after counter
- An interrupt is generated in interval timer mode

  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 b
- Choice of two resets

  Power-on reset and manual reset are available.

Figure 13.1 shows a block diagram of the WDT.



[Legend]
WTCSR: Watchdog timer control/status register
WTCNT: Watchdog timer counter

**Figure 13.1 Block Diagram of WDT**

### 13.3.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable register that increments on the selected clock. Whe
overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval t
mode. The WTCNT counter is not initialized by an internal reset due to the WDT overflo
WTCNT counter is initialized to H'00 only by a power-on reset using the $\overline{\text{RES}}$ pin. Use a
access to write to the WTCNT counter, with H'5A in the upper byte. Use a byte access to
WTCNT.

Note: WTCNT differs from other registers in that it is more difficult to write to. See sec
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 |

Note: WTCSR differs from other registers in that it is more difficult to write to. See se 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 t using the WDT to revoke software standby moc<br>0: Timer disabled: Count-up stops and WTCNT retained<br>1: Timer enabled |
| 6 | WT/IT | 0 | R/W | Timer Mode Select<br>Selects whether to use the WDT as a watchdog an interval timer.<br>0: Interval timer mode<br>1: Watchdog timer mode<br>Note: If WT/IT is modified when the WDT is op the up-count may not be performed corr |

| | | | | |
|---|---|---|---|---|
| | | | | 0: No overflow<br>1: WTCNT has overflowed in watchdog timer mo |
| 3 | IOVF | 0 | R/W | Interval Timer Overflow<br>Indicates that the WTCNT has overflowed in inte 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 WT count from the eight types obtainable by dividing peripheral clock (Pϕ). The overflow period that is inside the parenthesis in the table is the value wl 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 is operating, the up-count may not be pe correctly. Ensure that these bits are mod when the WDT is not operating. |

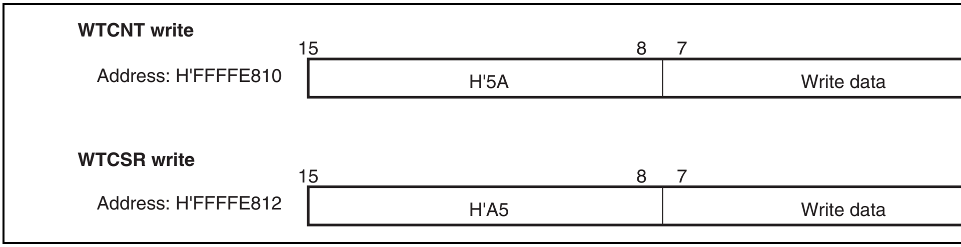


**Figure 13.2 Writing to WTCNT and WTCSR**

2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initial v
   the counter in the WTCNT counter. These values should ensure that the time till coun
   overflow is longer than the clock oscillation settling time.
3. Transition to software standby mode by executing a SLEEP instruction to stop the clo
4. The WDT starts counting by detecting a change in the level input to the NMI or IRQ p
5. When the WDT count overflows, the CPG starts supplying the clock and the LSI resu
   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 ever
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 co
   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 t
   counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1, asserts a
   through the $\overline{\text{WDTOVF}}$ pin for one cycle of the count clock specified by the CKS2 to C
   bits, and generates a reset of the type specified by the RSTS bit. The counter then resu
   counting.

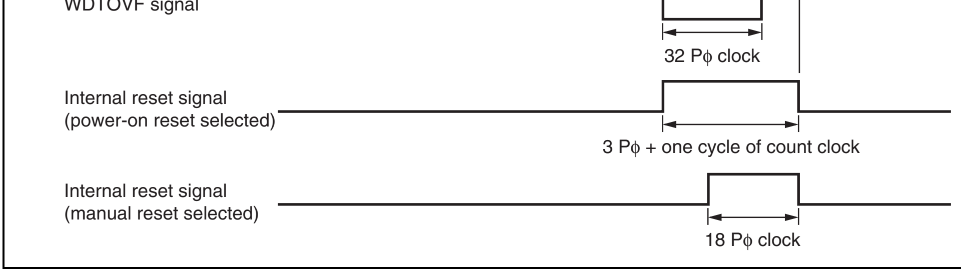


**Figure 13.3 Operation in Watchdog Timer Mode**
**(When WTCNT Count Clock is Specified to Pϕ/32 by CKS2 to CKS0)**

### 13.4.3 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every o
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 CKS
   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 int
   timer interrupt request is sent to the INTC. The counter then resumes counting.

| Name | Interrupt Source | Interrupt Enable Bit | Interrupt Flag Bit |
|---|---|---|---|
| ITI | Interval timer interrupt | — | Interval timer overflow fla |

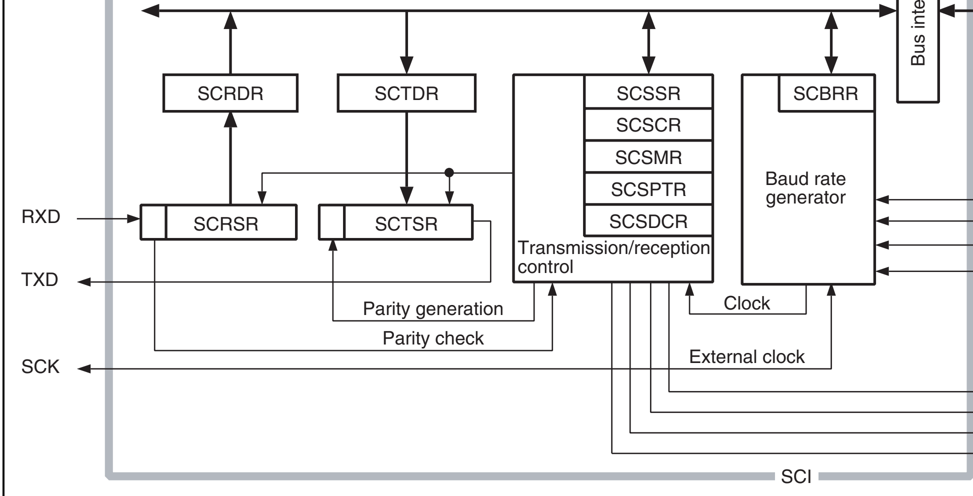
## 13.6 Usage Note

### 13.6.1 WTCNT Setting Value

If WTCNT is set to H'FF in interval timer mode, overflow does not occur when WTCNT from H'FF to H'00 after one cycle of count clock, but overflow occurs when WTCNT cha 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 chang H'FF to H'00 after one cycle of count clock.

- Choice of asynchronous or clock synchronous serial communication mode
- Asynchronous mode:
  - Serial data communication is performed by start-stop in character units. The SCI communicate with a universal asynchronous receiver/transmitter (UART), an asy communication interface adapter (ACIA), or any other communications chip that 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 error occurs.
- Clock synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCIF can co 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 independen SCI can transmit and receive simultaneously. Both sections use double buffering, so 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 (i clock) or SCK pin (external clock)
- Choice of LSB-first or MSB-first data transfer (except for 7-bit data in asynchronous



[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**

| | | | |
|---|---|---|---|
| | 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, o the channel designation.

| | | | | | | |
|---|---|---|---|---|---|---|
| | 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 |

### 14.3.2 Receive Data Register (SCRDR)

SCRDR is a register that stores serial receive data. After receiving one byte of serial dat transfers the received data from the receive shift register (SCRSR) into SCRDR for stor 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 co

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 registe 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 SCT SCTSR and starts transmitting again. If the TDRE flag in the serial status register (SCS to 1, the SCI does not transfer data from SCTDR to SCTSR. The CPU cannot read or w SCTSR directly.

| 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 select 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 asynchron 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 moc clock synchronous mode, the data length is alw eight bits, regardless of the CHR setting. Wher data is selected, the MSB (bit 7) of the transmi register is not transmitted.<br>0: 8-bit data<br>1: 7-bit data |

| | | | | |
|---|---|---|---|---|
| | | | | parity mode (O/E) setting. Receive d is checked according to the even/odd mode setting. |
| 4 | O/$\overline{\text{E}}$ | 0 | R/W | Parity mode<br>Selects even or odd parity when parity bits ar and checked. The O/$\overline{\text{E}}$ setting is used only in asynchronous mode and only when the parity (PE) is set to 1 to enable parity addition and c The O/$\overline{\text{E}}$ setting is ignored in clock synchrono or in asynchronous mode when parity additior checking is disabled.<br>0: Even parity<br>1: Odd parity<br>If even parity is selected, the parity bit is adde transmit data to make an even number of 1s i transmitted character and parity bit combined data is checked to see if it has an even numb the received character and parity bit combinec<br>If odd parity is selected, the parity bit is addec transmit data to make an odd number of 1s in transmitted character and parity bit combined data is checked to see if it has an odd numbe the received character and parity bit combinec |

| | | | | |
|---|---|---|---|---|
| | | | | bit is 1, it is treated as a stop bit, but if the seco<br>bit is 0, it is treated as the start bit of the next in<br>character.<br>Notes: 1. When transmitting, a single 1-bit is ad<br>the end of each transmitted character<br>2. When transmitting, two 1 bits are add<br>end of each transmitted character. |
| 2 | MP | 0 | R/W | Multiprocessor Mode (only in asynchronous mo<br>Enables or disables multiprocessor mode. The<br>O/$\overline{\text{E}}$ bit settings are ignored in multiprocessor m<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 b<br>generator. Four clock sources are available. Pϕ<br>Pϕ/16 and Pϕ/64. For further information on the<br>source, bit rate register settings, and baud rate<br>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 |

| Bit | Bit Name | Initial value | R/W | Description |
|---|---|---|---|---|
| 7 | TIE | 0 | R/W | Transmit Interrupt Enable<br>Enables or disables a transmit-data-empty int (TXI) to be issued when the TDRE flag in the status register (SCSSR) is set to 1 after seria data is sent from the transmit data register (S the transmit shift register (SCTSR).<br>TXI can be canceled by clearing the TDRE fla after reading TDRE = 1 or by clearing the TIE<br>0: Transmit-data-empty interrupt request (TXI disabled<br>1: Transmit-data-empty interrupt request (TXI enabled |
| 6 | RIE | 0 | R/W | Receive Interrupt Enable<br>Enables or disables a receive-data-full interru and a receive error interrupt (ERI) to be issue the RDRF flag in SCSSR is set to 1 after the s received is transferred from the receive shift r (SCRSR) to the receive data register (SCRDF<br>RXI can be canceled by clearing the RDRF fla reading RDRF =1. ERI can be canceled by cle FER, PER, or ORER flag to 0 after reading 1 flag. Both RXI and ERI can also be canceled clearing the RIE bit to 0.<br>0: Receive-data-full interrupt (RXI) and receiv interrupt (ERI) requests are disabled<br>1: Receive-data-full interrupt (RXI) and receiv interrupt (ERI) requests are enabled |

| | | | | |
|---|---|---|---|---|
| | | | | format in the serial mode register (S<br>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 flags (RDRF, FER, PER, and ORER flags retain their previous values.<br>2. Serial reception starts when a start detected in asynchronous mode, or synchronous clock input is detected synchronous mode. Select the rece format in SCSMR before setting RE |
| 3 | MPIE | 0 | R/W | Multiprocessor Interrupt Enable (only when MP SCSMR in asynchronous mode)<br>When this bit is set to 1, receive data in which multiprocessor bit is 0 is skipped and setting of RDRF, FER, and ORER status flags in SCSSF prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically to 0 and normal receiving operation is resumed details, refer to section 14.4.4, Multiprocessor Communication Function. |

| | | | | |
|---|---|---|---|---|
| 1, 0 | CKE[1:0] | 00 | R/W | Clock Enable 1 and 0<br>Select the SCI clock source and enable or dis output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin used for serial clock output or serial clock inpu<br>When selecting the clock output in clock sync mode, set the C/$\overline{\text{A}}$ bit in SCSMR to 1 and ther CKE1 and CKE0. For details on clock source refer to table 14.14 in section 14.4, Operation<br>• Asynchronous mode<br>00: Internal clock, SCK pin used for input pin signal is ignored.)<br>01: Internal clock, SCK pin used for clock outp<br>10: External clock, SCK pin used for clock inp<br>11: External clock, SCK pin used for clock inp<br>• Clock synchronous mode<br>00: Internal clock, SCK pin used for synchron output<br>01: Internal clock, SCK pin used for synchron output<br>10: External clock, SCK pin used for synchror input<br>11: External clock, SCK pin used for synchror input<br>Notes: 1. The output clock frequency is 16 ti bit rate.<br>2. The input clock frequency is 16 tim rate. |

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 transmit data register (SCTDR) to the transm register (SCTSR) and SCTDR has become r be written with next serial transmit data.<br>0: Indicates that SCTDR holds valid transmit<br>[Clearing conditions]<br>• When 0 is written to TDRE after reading<br>• When the DTC is activated by a TXI inter transmit data is transferred to SCTDR wh DISEL bit of MRB in the DTC is 0<br>1: Indicates that SCTDR does not hold valid 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 and data can be written to SCTDR |

- When the DTC is activated by an RXI in
  and data is transferred from SCRDR wh
  DISEL bit of MRB in the DTC is 0

1: Indicates that valid received data is store
   SCRDR

[Setting condition]

- When serial reception ends normally an
  data is transferred from SCRSR to SCR

Note: SCRDR and the RDRF flag are not af
retain their previous states even if an
detected during data reception or if th
the serial control register (SCSCR) is
0. If reception of the next data is com
while the RDRF flag is still set to 1, a
error will occur and the received data
lost.

1: Indicates that an overrun error occurred du
reception*[2]

[Setting condition]

- When the next serial reception is complet
  RDRF = 1

Notes: 1. The ORER flag is not affected an
its previous value when the RE bi
SCSCR is cleared to 0.

2. The receive data prior to the over
is retained in SCRDR, and the da
received subsequently is lost. Sul
serial reception cannot be continu
the ORER flag is set to 1.

1: Indicates that a framing error occurred du
reception

[Setting condition]

- When the SCI founds that the stop bit at of the received data is 0 after completing reception*[2]

Notes: 1. The FER flag is not affected and its previous value when the RE SCSCR is cleared to 0.

2. In 2-stop-bit mode, only the first checked for a value to 1; the sec bit is not checked. If a framing e occurs, the receive data is trans SCRDR but the RDRF flag is no Subsequent serial reception car continued while the FER flag is s

1: Indicates that a parity error occurred durin
reception*[2]

[Setting condition]

- When the number of 1s in the received da parity does not match the even or odd pa specified by the $O/\overline{E}$ bit in the serial mode (SCSMR).

Notes: 1. The PER flag is not affected and its previous value when the RE bi SCSCR is cleared to 0.

2. If a parity error occurs, the receiv transferred to SCRDR but the RD is not set. Subsequent serial rece cannot be continued while the PE set to 1.

| | | | | |
|---|---|---|---|---|
| | | | | 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<br>of a 1-byte serial transmit character<br>Note: The TEND flag value becomes unde<br>data is written to SCTDR by activatin<br>by a TXI interrupt. In this case, do no<br>TEND flag as the transmit end flag. |
| 1 | MPB | 0 | R | Multiprocessor Bit<br>Stores the multiprocessor bit found in the re<br>data. When the RE bit in SCSCR is cleared<br>previous state is retained. |
| 0 | MPBT | 0 | R/W | Multiprocessor Bit Transfer<br>Specifies the multiprocessor bit value to be<br>the transmit frame. |

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

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 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 E set to 1, the SCI does not request an RXI inter the CPU even if the RIE bit is set to 1.<br>0: The RIE bit enables or disables RXI and ER interrupts. While the RIE bit is 1, RXI and EF interrupts are sent to the INTC.<br>1: While the RIE bit is 1, only the ERI interrupt the INTC. |
| 6 to 4 | — | All 0 | — | Reserved<br>These bits are always read as 0. The write valu always be 0. |
| 3 | SPB1IO | 0 | R/W | Clock Port Input/Output in Serial Port<br>Specifies the input/output direction of the SCK serial port. To output the data specified in the S bit through the SCK pin as a port output pin, se bit in SCSMR and the CKE1 and CKE0 bits in to 0.<br>0: Does not output the SPB1DT bit value throu SCK pin.<br>1: Outputs the SPB1DT bit value through the S |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | Together with the SPB0DT bit and the TE bit<br>SCSCR, controls the TXD pin. |
| 0 | SPB0DT | 1 | R/W | Serial Port Break Data<br>Together with the SPB0IO bit and TE bit in SC controls the TXD pin. Note that the TXD pin fu needs to have been selected with the pin fun controller (PFC). |

| TE bit setting in SCSCR | SPB0IO bit setting | SPB0DT bit setting | State of T |
|---|---|---|---|
| 0 | 0 | * | SPB0DT c disabled (initial stat |
| 0 | 1 | 0 | Output, lo |
| 0 | 1 | 1 | Output, hi |
| 1 | * | * | Output for data in ac the serial logic |

Note: * Don't care

| 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 valu always be 1. |
| 3 | DIR | 0 | R/W | Data Transfer Direction<br>Selects the serial/parallel conversion format. V an 8-bit transmit/receive format.<br>0: SCTDR contents are transmitted in LSB-firs Receive data is stored in SCRDR in LSB-firs<br>1: SCTDR contents are transmitted in MSB-firs Receive data is stored in SCRDR in MSB-fir |
| 2 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value sh always be 0. |
| 1 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value sh always be 1. |
| 0 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value sh always be 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 n, see table 14.3.)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Tables 14.4 to 14.6 show examples of SCBRR settings in asynchronous mode, and tables
14.9 show examples of SCBRR settings in clock synchronous mode.

**Table 14.4 Bit Rates and SCBRR Settings in Asynchronous Mode (1)**

| | Pφ (MHz) | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | | | 12 | | | 14 | | | 16 | | | 18 | | | | |
| Bit Rate (bits/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N |
| 110 | 2 | 177 | -0.25 | 2 | 212 | 0.03 | 2 | 248 | -0.17 | 3 | 70 | 0.03 | 3 | 79 | -0.12 | 3 | 88 |
| 150 | 2 | 129 | 0.16 | 2 | 155 | 0.16 | 2 | 181 | 0.16 | 2 | 207 | 0.16 | 2 | 233 | 0.16 | 3 | 64 |
| 300 | 2 | 64 | 0.16 | 2 | 77 | 0.16 | 2 | 90 | 0.16 | 2 | 103 | 0.16 | 2 | 116 | 0.16 | 2 | 12 |
| 600 | 1 | 129 | 0.16 | 1 | 155 | 0.16 | 1 | 181 | 0.16 | 1 | 207 | 0.16 | 1 | 233 | 0.16 | 2 | 64 |
| 1200 | 1 | 64 | 0.16 | 1 | 77 | 0.16 | 1 | 90 | 0.16 | 1 | 103 | 0.16 | 1 | 116 | 0.16 | 1 | 12 |
| 2400 | 0 | 129 | 0.16 | 0 | 155 | 0.16 | 0 | 181 | 0.16 | 0 | 207 | 0.16 | 0 | 233 | 0.16 | 1 | 64 |
| 4800 | 0 | 64 | 0.16 | 0 | 77 | 0.16 | 0 | 90 | 0.16 | 0 | 103 | 0.16 | 0 | 116 | 0.16 | 0 | 12 |
| 9600 | 0 | 32 | -1.36 | 0 | 38 | 0.16 | 0 | 45 | -0.93 | 0 | 51 | 0.16 | 0 | 58 | -0.69 | 0 | 64 |
| 14400 | 0 | 21 | -1.36 | 0 | 25 | 0.16 | 0 | 29 | 1.27 | 0 | 34 | -0.79 | 0 | 38 | 0.16 | 0 | 42 |
| 19200 | 0 | 15 | 1.73 | 0 | 19 | -2.34 | 0 | 22 | -0.93 | 0 | 25 | 0.16 | 0 | 28 | 1.02 | 0 | 32 |
| 28800 | 0 | 10 | -1.36 | 0 | 12 | 0.16 | 0 | 14 | 1.27 | 0 | 16 | 2.12 | 0 | 19 | -2.34 | 0 | 21 |
| 31250 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 13 | 0.00 | 0 | 15 | 0.00 | 0 | 17 | 0.00 | 0 | 19 |
| 38400 | 0 | 7 | 1.73 | 0 | 9 | -2.34 | 0 | 10 | 3.57 | 0 | 12 | 0.16 | 0 | 14 | -2.34 | 0 | 15 |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1200 | 1 | 142 | 0.16 | 1 | 155 | 0.16 | 1 | 168 | 0.16 | 1 | 181 | 0.16 | 1 | 194 | 0.16 | 1 | 2 |
| 2400 | 1 | 71 | -0.54 | 1 | 77 | 0.16 | 1 | 84 | -0.43 | 1 | 90 | 0.16 | 1 | 97 | -0.35 | 1 | 1 |
| 4800 | 0 | 142 | 0.16 | 0 | 155 | 0.16 | 0 | 168 | 0.16 | 0 | 181 | 0.16 | 0 | 194 | 0.16 | 0 | 2 |
| 9600 | 0 | 71 | -0.54 | 0 | 77 | 0.16 | 0 | 84 | -0.43 | 0 | 90 | 0.16 | 0 | 97 | -0.35 | 0 | 1 |
| 14400 | 0 | 47 | -0.54 | 0 | 51 | 0.16 | 0 | 55 | 0.76 | 0 | 60 | -0.39 | 0 | 64 | 0.16 | 0 | 6 |
| 19200 | 0 | 35 | -0.54 | 0 | 38 | 0.16 | 0 | 41 | 0.76 | 0 | 45 | -0.93 | 0 | 48 | -0.35 | 0 | 5 |
| 28800 | 0 | 23 | -0.54 | 0 | 25 | 0.16 | 0 | 27 | 0.76 | 0 | 29 | 1.27 | 0 | 32 | -1.36 | 0 | 3 |
| 31250 | 0 | 21 | 0.00 | 0 | 23 | 0.00 | 0 | 25 | 0.00 | 0 | 27 | 0.00 | 0 | 29 | 0.00 | 0 | 3 |
| 38400 | 0 | 17 | -0.54 | 0 | 19 | -2.34 | 0 | 20 | 0.76 | 0 | 22 | -0.93 | 0 | 23 | 1.73 | 0 | 2 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1200 | 1 | 220 | 0.16 | 1 | 233 | 0.16 | 1 | 246 | 0.16 | 2 | 64 |
| 2400 | 1 | 110 | -0.29 | 1 | 116 | 0.16 | 1 | 123 | -0.24 | 1 | 129 |
| 4800 | 0 | 220 | 0.16 | 0 | 233 | 0.16 | 0 | 246 | 0.16 | 1 | 64 |
| 9600 | 0 | 110 | -0.29 | 0 | 116 | 0.16 | 0 | 123 | -0.24 | 0 | 129 |
| 14400 | 0 | 73 | -0.29 | 0 | 77 | 0.16 | 0 | 81 | 0.57 | 0 | 86 |
| 19200 | 0 | 54 | 0.62 | 0 | 58 | -0.69 | 0 | 61 | -0.24 | 0 | 64 |
| 28800 | 0 | 36 | -0.29 | 0 | 38 | 0.16 | 0 | 40 | 0.57 | 0 | 42 |
| 31250 | 0 | 33 | 0.00 | 0 | 35 | 0.00 | 0 | 37 | 0.00 | 0 | 39 |
| 38400 | 0 | 27 | -1.18 | 0 | 28 | 1.02 | 0 | 30 | -0.24 | 0 | 32 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10000 | 0 | 249 | 1 | 74 | 1 | 87 | 1 | 99 | 1 | 112 | 1 |
| 25000 | 0 | 99 | 0 | 119 | 0 | 139 | 0 | 159 | 0 | 179 | 0 |
| 50000 | 0 | 49 | 0 | 59 | 0 | 69 | 0 | 79 | 0 | 89 | 0 |
| 100000 | 0 | 24 | 0 | 29 | 0 | 34 | 0 | 39 | 0 | 44 | 0 |
| 250000 | 0 | 9 | 0 | 11 | 0 | 13 | 0 | 15 | 0 | 17 | 0 |
| 500000 | 0 | 4 | 0 | 5 | 0 | 6 | 0 | 7 | 0 | 8 | 0 |
| 1000000 | — | — | 0 | 2 | — | — | 0 | 3 | — | — | 0 |
| 2500000 | 0 | 0* | — | — | — | — | — | — | — | — | 0 |
| 5000000 | | | — | — | — | — | — | — | — | — | 0 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10000 | 1 | 137 | 1 | 149 | 1 | 162 | 1 | 174 | 1 | 187 | 1 |
| 25000 | 0 | 219 | 0 | 239 | 1 | 64 | 1 | 69 | 1 | 74 | 1 |
| 50000 | 0 | 109 | 0 | 119 | 0 | 129 | 0 | 139 | 0 | 149 | 0 |
| 100000 | 0 | 54 | 0 | 59 | 0 | 64 | 0 | 69 | 0 | 74 | 0 |
| 250000 | 0 | 21 | 0 | 23 | 0 | 25 | 0 | 27 | 0 | 29 | 0 |
| 500000 | 0 | 10 | 0 | 11 | 0 | 12 | 0 | 13 | 0 | 14 | 0 |
| 1000000 | — | — | 0 | 5 | — | — | 0 | 6 | — | — | 0 |
| 2500000 | — | — | — | — | — | — | — | — | 0 | 2 | — |
| 5000000 | — | — | — | — | — | — | — | — | — | — | — |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10000 | 1 | 212 | 1 | 224 | 1 | 237 | 1 |
| 25000 | 1 | 84 | 1 | 89 | 1 | 94 | 1 |
| 50000 | 0 | 169 | 0 | 179 | 0 | 189 | 0 |
| 100000 | 0 | 84 | 0 | 89 | 0 | 94 | 0 |
| 250000 | 0 | 33 | 0 | 35 | 0 | 37 | 0 |
| 500000 | 0 | 16 | 0 | 17 | 0 | 18 | 0 |
| 1000000 | — | — | 0 | 8 | — | — | 0 |
| 2500000 | — | — | — | — | — | — | 0 |
| 5000000 | — | — | — | — | — | — | 0 |

[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.

| | | | |
|---|---|---|---|
| 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 |

|  |  |  |
|---|---|---|
| 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 |

|  |  |  |
|---|---|---|
| 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 |

(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 selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and
- 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 t chip baud rate generator and can output a clock with a frequency 16 times the bit
  - When an external clock is selected, the external clock input must have a frequenc 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 g and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. Th baud rate generator is not used.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | 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/$\overline{\text{A}}$ | 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 th is output. |
| | 1 | 0 | | External | Input a clock with frequency 16 tim bit rate. |
| | | 1 | | | |
| 1 | 0 | 0 | Clock synchronous | Internal | Serial clock is output. |
| | | 1 | | | |
| | 1 | 0 | | External | Input the serial clock. |
| | | 1 | | | |

monitors the line and starts serial communication when the line goes to the space (low) s
indicating a start bit. One serial character consists of a start bit (low), data (LSB first), p
(high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the s
The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times t
Receive data is latched at the center of each bit.

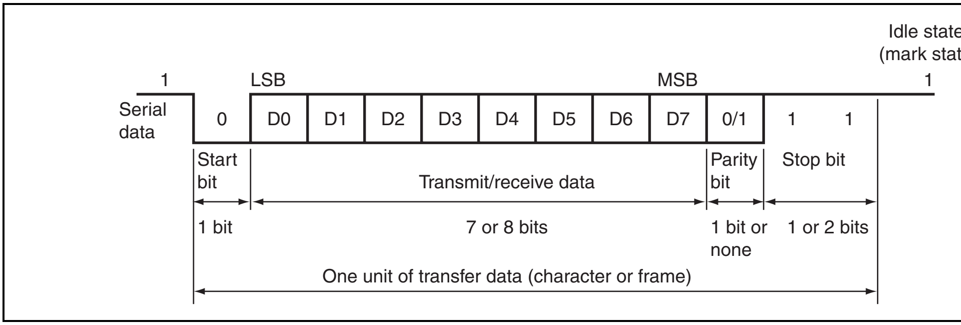


**Figure 14.2 Example of Data Format in Asynchronous Communication**
**(8-Bit Data with Parity and Two Stop Bits)**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 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 | S |
| 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 | |
| 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

frequency of this output clock is equal to 16 times the desired bit rate.

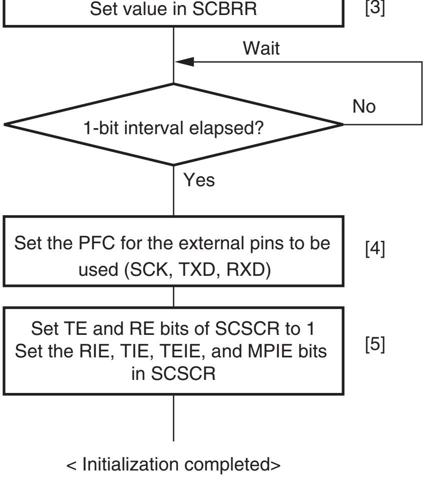
### (3) Transmitting and Receiving Data

**SCI Initialization (Asynchronous Mode):**

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control regis (SCSCR), then initialize the SCI as follows.

When changing the operation mode or the communication format, always clear the TE a to 0 before following the procedure given below. Clearing the TE bit to 0 sets the TDRE and initializes the transmit shift register (SCTSR). Clearing the RE bit to 0, however, do initialize the RDRF, PER, FER, and ORER flags or receive data register (SCRDR), whi their previous contents.

When an external clock is used, the clock should not be stopped during initialization or operation. SCI operation becomes unreliable if the clock is stopped.


Set value in SCBRR
[3]
Wait
1-bit interval elapsed?
No
Yes
Set the PFC for the external pins to be used (SCK, TXD, RXD)
[4]
Set TE and RE bits of SCSCR to 1
Set the RIE, TIE, TEIE, and MPIE bits in SCSCR
[5]
< Initialization completed>

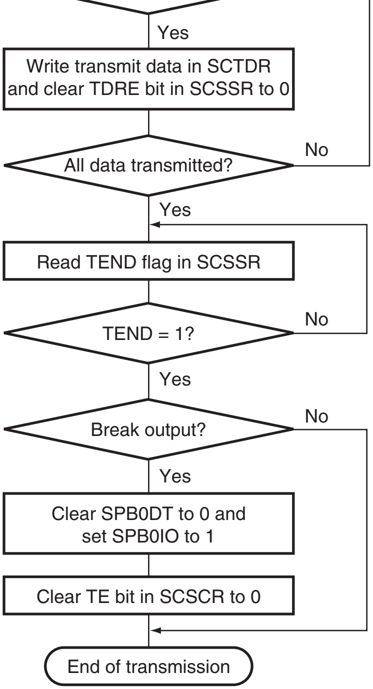
setting.

[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**

the TDRE flag to 0.


Yes
Write transmit data in SCTDR and clear TDRE bit in SCSSR to 0
All data transmitted?
No
Yes
Read TEND flag in SCSSR
TEND = 1?
No
Yes
Break output?
No
Yes
Clear SPB0DT to 0 and set SPB0IO to 1
Clear TE bit in SCSCR to 0
End of transmission

[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**

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 multipro bit is output. (A format in which neither parity nor multiprocessor bit is output car 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 transmiss 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 se 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 the "mark state" is entered in which 1 is output. If the TEIE bit in SCSCR is set to 1 at thi TEI interrupt request is generated.

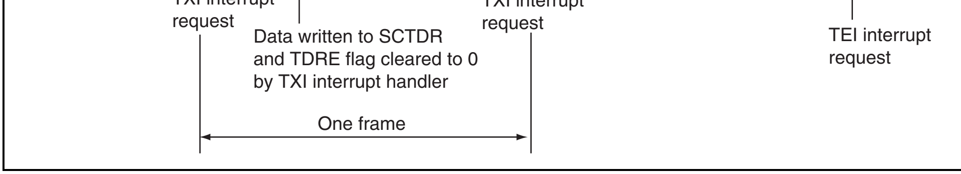
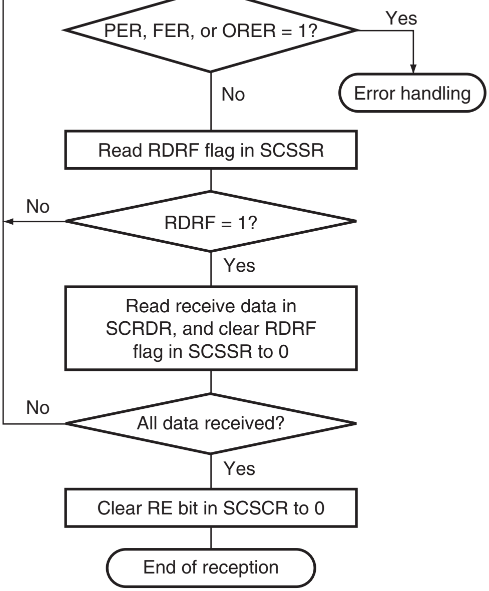


**Figure 14.5 Example of Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)**



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)**

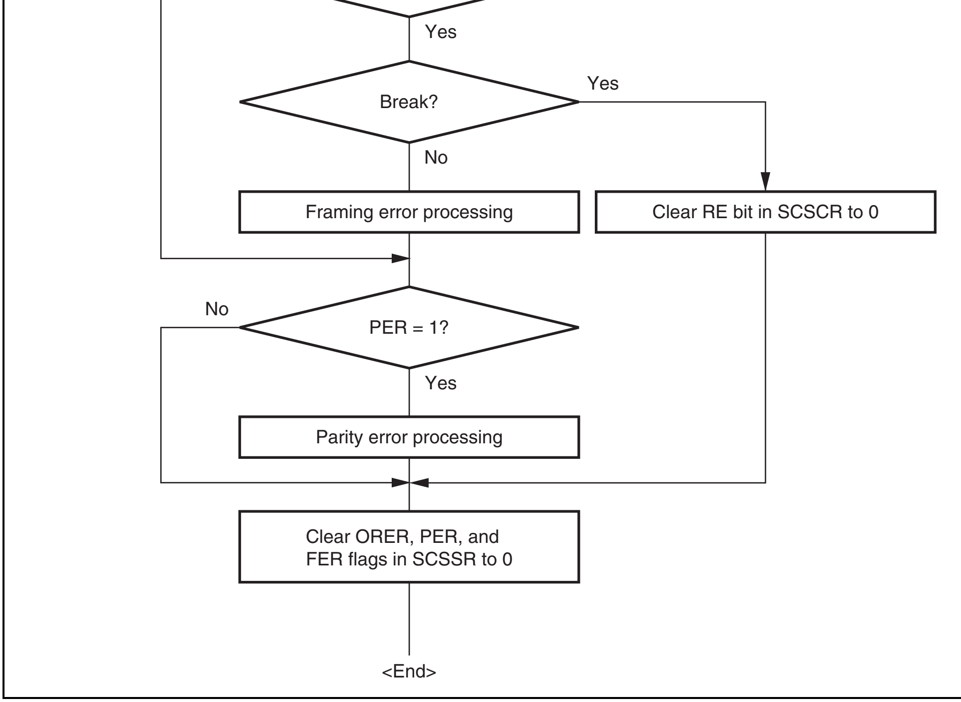


Figure 14.7 Sample Flowchart for Receiving Serial Data (2)

B. Stop bit check: The SCI checks whether the stop bit is 1. If there are two stop bits, first is checked.

C. Status check: The SCI checks whether the RDRF flag is 0 and the received data ca 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 st 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 ad the RDRF flag will not be set to 1 after reception; be sure to clear the error fla

4. If the EIO bit in SCSPTR is cleared to 0 and the RIE bit in SCSCR is set to 1 when th flag changes to 1, a receive-data-full interrupt (RXI) request is generated. If the RIE b SCSCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive error int (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 r transferred from SCR SCRDR. |
| Framing error | FER | When the stop bit is 0 | The received data is transferred from SCR 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 SCR SCRDR. |

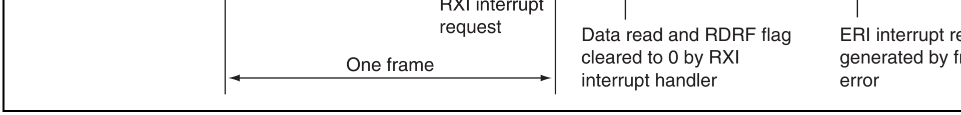


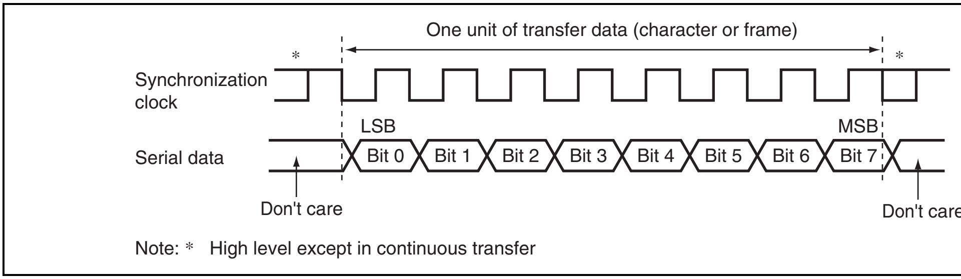
**Figure 14.8 Example of SCI Receive Operation (8-Bit Data, Parity, One Stop Bit)**

### 14.4.3 Clock Synchronous Mode

In clock synchronous mode, the SCIF transmits and receives data in synchronization wit pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full-duplex communication is poss sharing the same clock. Both the transmitter and receiver have a double-buffered structu data can be read or written during transmission or reception, enabling continuous data tr

Figure 14.9 shows the general format in clock synchronous serial communication.



Note: * High level except in continuous transfer

**Figure 14.9 Data Format in Clock Synchronous Communication**

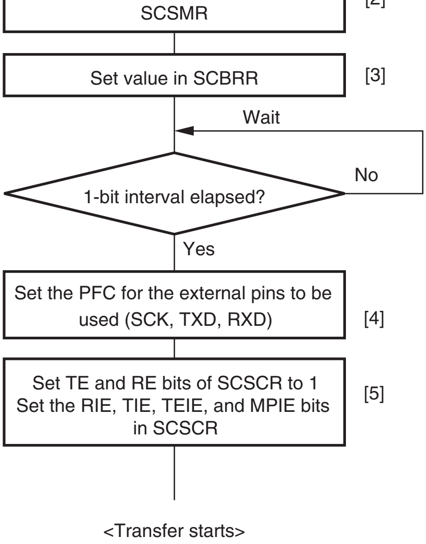
### (2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input SCK pin can be selected as the SCI transmit/receive clock. For selection of the SCI clock see table 14.14.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Ei pulses are output per transmitted or received character. When the SCI is not transmitting receiving, the clock signal remains in the high state. When only reception is performed, o the synchronous clock continues until an overrun error occurs or the RE bit is cleared to reception of n characters, select the external clock as the clock source. If the internal cloc be used, set RE and TE to 1, then transmit n characters of dummy data at the same time a receiving the n characters of data.

### (3) Transmitting and Receiving Data

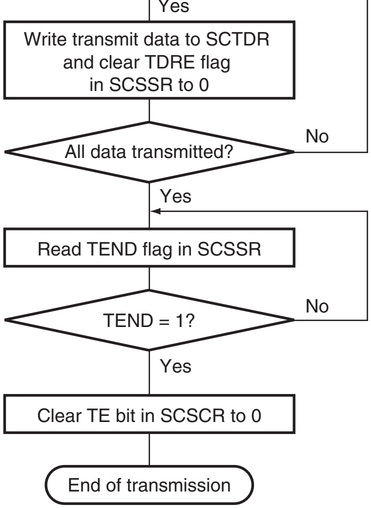
**SCI Initialization (Clock Synchronous Mode):** Before transmitting, receiving, or chang mode or communication format, the software must clear the TE and RE bits to 0 in the se control register (SCSCR), then initialize the SCI. Clearing TE to 0 sets the TDRE flag to initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initial RDRF, PER, FER, and ORER flags and receive data register (SCRDR), which retain thei previous contents.


[2]
SCSMR
Set value in SCBRR
[3]
Wait
No
1-bit interval elapsed?
Yes
Set the PFC for the external pins to be used (SCK, TXD, RXD)
[4]
Set TE and RE bits of SCSCR to 1
Set the RIE, TIE, TEIE, and MPIE bits in SCSCR
[5]
<Transfer starts>

[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 p is in a mark state during transmitting. Whe 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**


Yes
Write transmit data to SCTDR and clear TDRE flag in SCSSR to 0
All data transmitted?
No
Yes
Read TEND flag in SCSSR
TEND = 1?
No
Yes
Clear TE bit in SCSCR to 0
End of transmission

[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**

3. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7). If the TDR 0, the data is transferred from SCTDR to SCTSR and serial transmission of the next started, If the TDRE flag is 1, the TEND flag in SCSSR is set to 1, the MSB (bit 7) i 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.

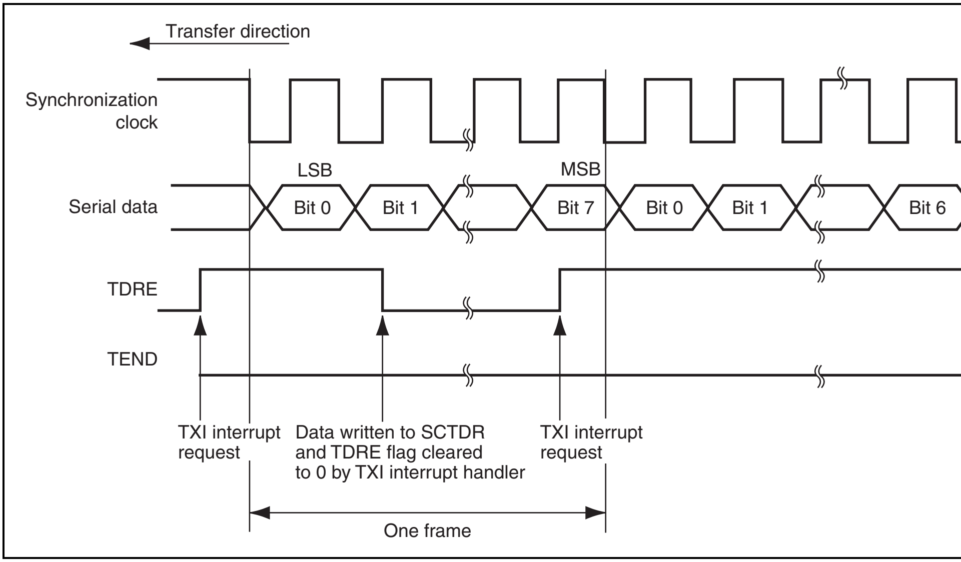


**Figure 14.12 Example of SCI Transmit Operation**

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.

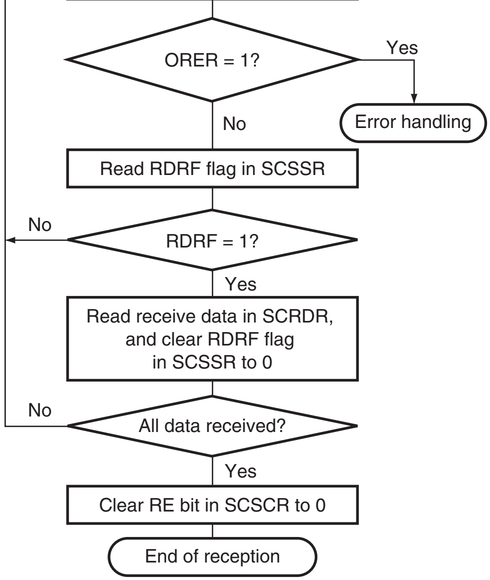


**Figure 14.13 Sample Flowchart for Receiving Serial Data (1)**

**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 receivi data, the SCI checks whether the RDRF flag is 0 and the receive data can be transfer SCRSR to SCRDR. If this check is passed, the SCI sets the RDRF flag to 1 and store received data in SCRDR. If a receive error is detected, the SCI operates as shown in 14.16. In this state, subsequent reception cannot be continued. In addition, the RDRF 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 SCSCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set t RIE bit in SCSCR is also set to 1, the SCI requests a receive error interrupt (ERI).

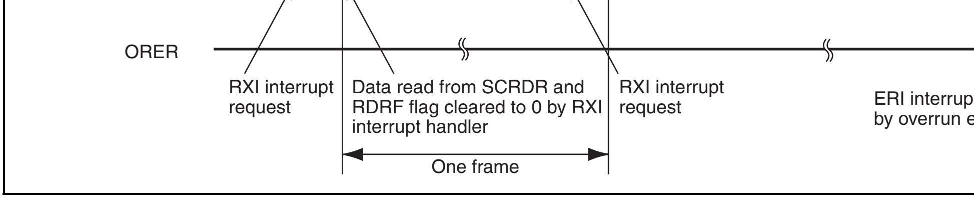
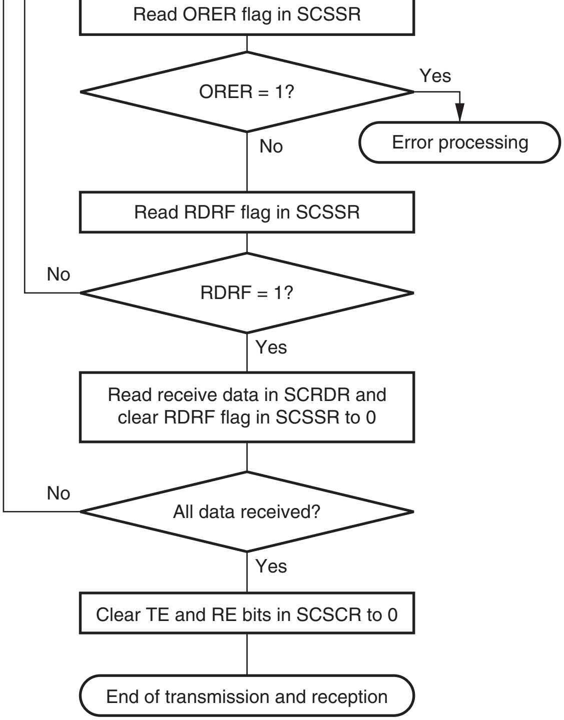


**Figure 14.15 Example of SCI Receive Operation**

**Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)**
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
transmission and reception.


Read ORER flag in SCSSR
ORER = 1?
Yes
Error processing
No
Read RDRF flag in SCSSR
No
RDRF = 1?
Yes
Read receive data in SCRDR and clear RDRF flag in SCSSR to 0
No
All data received?
Yes
Clear TE and RE bits in SCSCR to 0
End of transmission and reception

[3] SCI status check and receive data re
Read SCSSR and check that the RD
set to 1, then read the receive data i
and clear the RDRF flag to 0. Trans
RDRF flag from 0 to 1 can also be id
an RXI interrupt.

[4] Serial transmission/reception continu
procedure:
To continue serial transmission/recep
before the MSB (bit 7) of the current
received, finish reading the RDRF fla
SCRDR, and clearing the RDRF flag
before the MSB (bit 7) of the current
transmitted, read 1 from the TDRE fl
confirm that writing is possible. Ther
to SCTDR and clear the TDRE flag t
Checking and clearing of the TDRE f
automatic when the DTC is activated
transmit data empty interrupt (TXI) re
data is written to SCTDR. Also, the
is cleared automatically when the DT
activated by a receive data full interru
request and the SCRDR value is rea

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, firs
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**

inter-processor communication using the multiprocessor format. The transmitting station sends the ID code of the receiving station with which it wants to perform serial communi data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multip bit added. The receiving station skips data until data with a 1 multiprocessor bit is sent. W with a 1 multiprocessor bit is received, the receiving station compares that data with its o The station whose ID matches then receives the data sent next. Stations whose ID does n 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 s transfer of receive data from SCRSR to SCRDR, error flag detection, and setting the SCS flags, RDRF, FER, and OER to 1 are inhibited until data with a 1 multiprocessor bit is re On reception of receive character with a 1 multiprocessor bit, the MPBR bit in SCSSR is and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE b 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 s are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

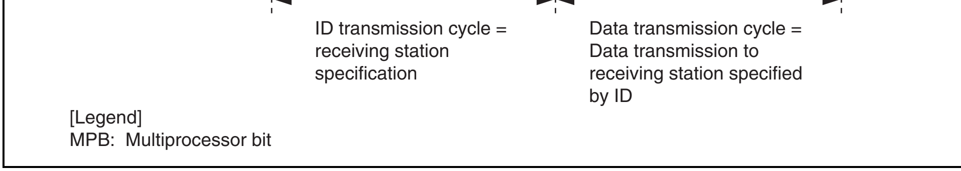
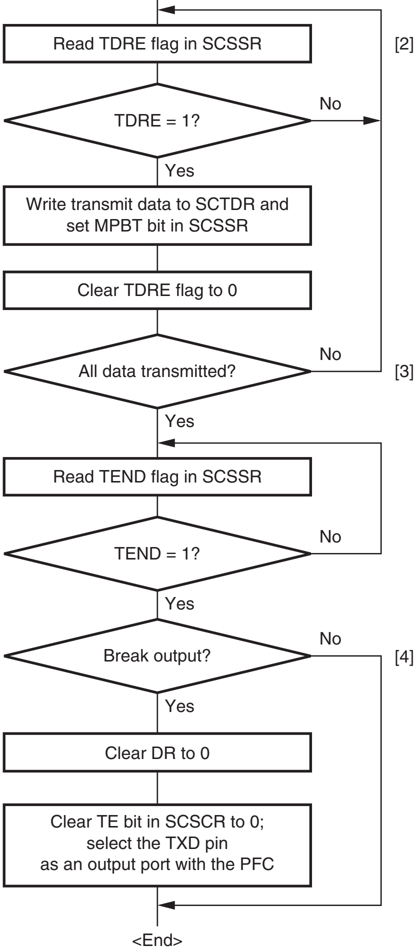


**Figure 14.17 Example of Communication Using Multiprocessor Format
(Transmission of Data H'AA to Receiving Station A)**



data is not transmitted.

[2] SCI status check and transmit data write: Read SCSSR and check that the TDRE flag is s to 1, then write transmit data to SCTDR. Set th MPBT bit in SCSSR to 0 or 1. Finally, clear the TDRE flag to 0. To transmit an ID after the SCI is initialized, wri the ID to SCTDR. The data is immediately transferred to SCTSR and the TDRE flag is set 1. At this point the ID has not yet been transmit from the TXD pin, so it is necessary to maintain the MPBT value at 1. Clear the MPBT bit to 0 a 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 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 clearin of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TX request, and data is written to SCTDR.

[4] Break output at the end of serial transmission: To output a break in serial transmission, first cle the port data register (DR) to 0, then clear the T 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**

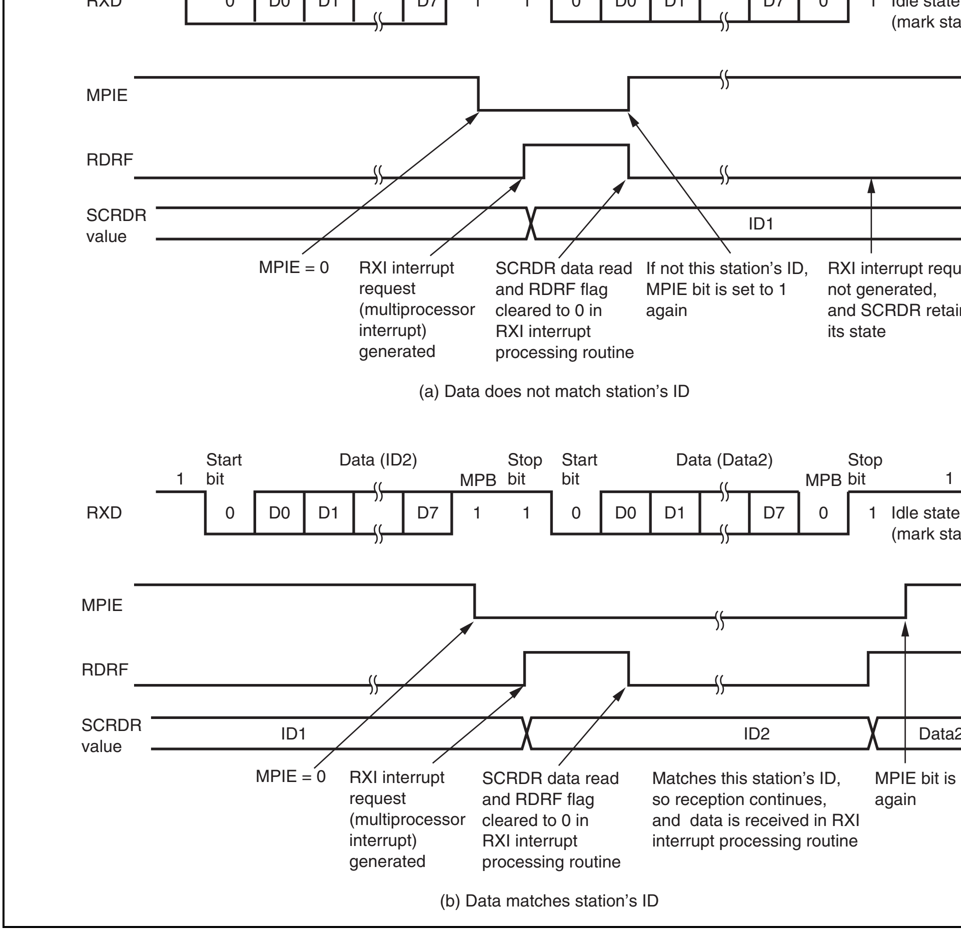
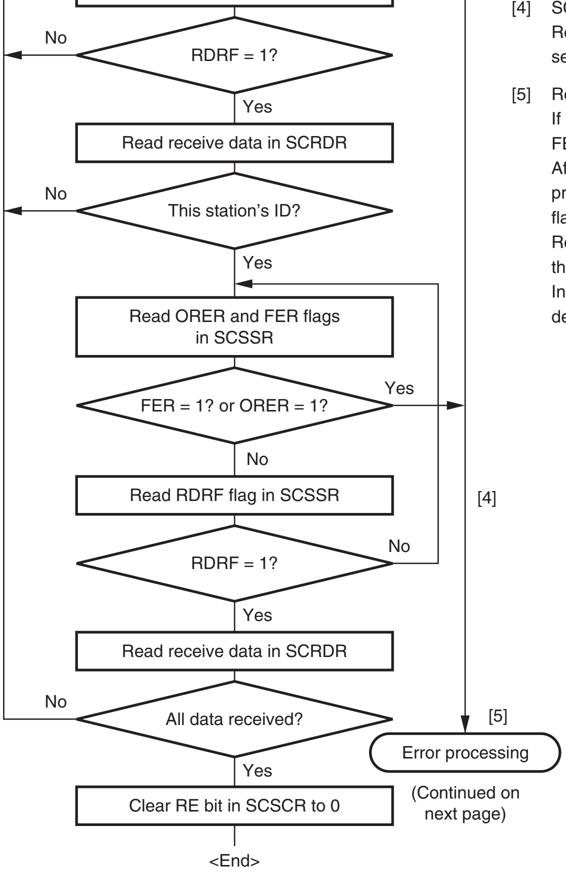

RXD
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idle state
(mark sta
MPIE
RDRF
SCRDR
value
ID1
MPIE = 0
RXI interrupt request (multiprocessor interrupt) generated
SCRDR data read and RDRF flag cleared to 0 in RXI interrupt processing routine
If not this station's ID, MPIE bit is set to 1 again
RXI interrupt requ not generated, and SCRDR retai its state
(a) Data does not match station's ID
1
Start bit
Data (ID2)
MPB
Stop bit
Start bit
Data (Data2)
MPB
Stop bit
1
RXD
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idle state
(mark sta
MPIE
RDRF
SCRDR
value
ID1
ID2
Data2
MPIE = 0
RXI interrupt request (multiprocessor interrupt) generated
SCRDR data read and RDRF flag cleared to 0 in RXI interrupt processing routine
Matches this station's ID, so reception continues, and data is received in RXI interrupt processing routine
MPIE bit is again
(b) Data matches station's ID

**Figure 14.19 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)**



[4] SCI status check and data reception: Read SCSSR and check that the RDRF fla set to 1, then read the data in SCRDR.

[5] Receive error processing and break detecti If a receive error occurs, read the ORER ar FER flags in SCSSR to identify the error. After performing the appropriate error processing, ensure that the ORER and FEF 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 l detected by reading the RXD pin value.

**Figure 14.20 Sample Multiprocessor Serial Reception Flowchart (1)**

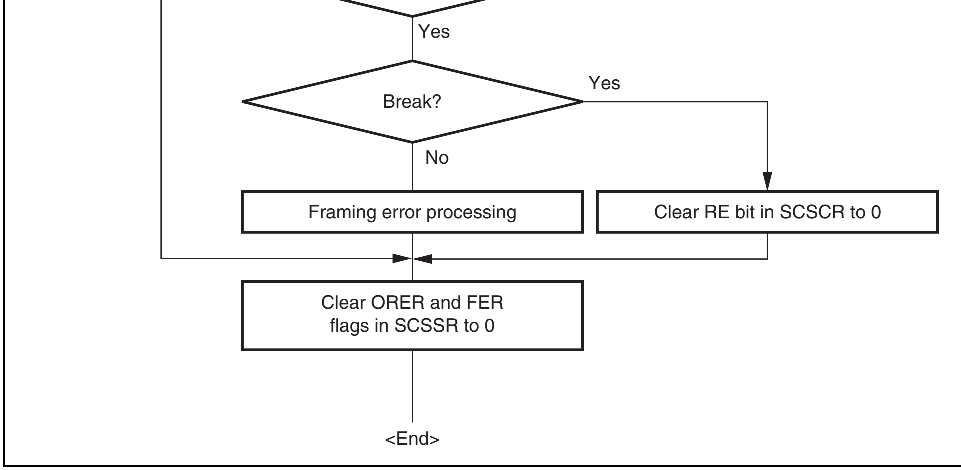

Yes
Break?
Yes
No
Framing error processing
Clear RE bit in SCSCR to 0
Clear ORER and FER flags in SCSSR to 0
<End>

Figure 14.21 Sample Multiprocessor Serial Reception Flowchart (2)

transfer data. The TDRE flag is automatically cleared to 0 when data is written to the tran register (SCTDR) through the DTC.

When the RDRF flag in SCSSR is set to 1, an RDR full interrupt request is generated. Th can be used to activate the DTC to transfer data. The RDRF flag is automatically cleared when data is read from the receive data register (SCRDR) through the DTC.

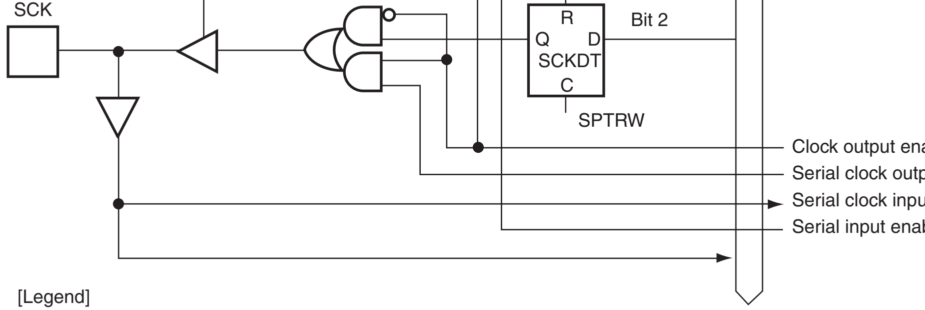
When the ORER, FER, or PER flag in SCSSR is set to 1, an ERI interrupt request is gene This request cannot be used to activate the DTC. It is possible to disable generation of RX interrupt requests and allow only ERI interrupt requests to be generated during data recep processing. To accomplish this, set the RIE bit to 1 and the EIO bit in SCSPTR to 1. Note setting the EIO bit to 1 will prevent the DTC from transferring received data because no R interrupt requests are generated.

When the TEND flag in SCSSR is set to 1, a TEI interrupt request is generated. This requ cannot be used to activate the DTC.

The TXI interrupt indicates that transmit data can be written, and the TEI interrupt indica 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 |



[Legend]
SPTRW: SCSPTR write

Note: * These signals control the SCK pin according to the settings of the C/$\overline{\text{A}}$ bit in SCSMR and bits CKE1 and CKE0 in SCSCR.

**Figure 14.22 SCKIO Bit, SCKDT Bit, and SCK Pin**

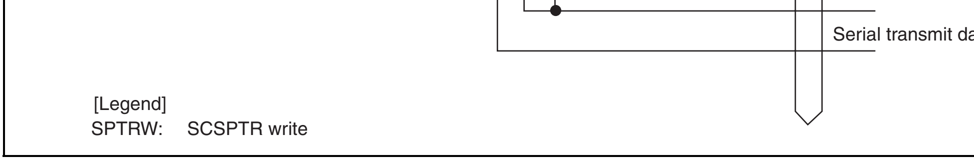


Figure 14.23 SPBIO Bit, SPBDT Bit, and TXD Pin

lost because the data has not yet been transferred to SCTSR. Before writing transmit dat 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 sho table 14.18. When an overrun error occurs, data is not transferred from the receive shift (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**

| | SCSSR Status Flags | | | | Recei Trans SCRS |
|---|---|---|---|---|---|
| **Receive Errors Generated** | **RDRF** | **ORER** | **FER** | **PER** | **SCRD** |
| Overrun error | 1 | 1 | 0 | 0 | Not tra |
| Framing error | 0 | 0 | 1 | 0 | Transf |
| Parity error | 0 | 0 | 0 | 1 | Transf |
| Overrun error + framing error | 1 | 1 | 1 | 0 | Not tra |
| Overrun error + parity error | 1 | 1 | 0 | 1 | Not tra |
| Framing error + parity error | 0 | 0 | 1 | 1 | Transf |
| Overrun error + framing error + parity error | 1 | 1 | 1 | 1 | Not tra |

Until TE bit is set to 1 (enabling transmission) after initializing, TXD pin does not work. the period, mark status is performed by SPB0DT bit. Therefore, the SPB0IO and SPB0D should be set to 1 (high level output).

To send a break signal during serial transmission, clear the SPB0DT bit to 0 (low level), the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is i 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 asynch mode. In reception, the SCI synchronizes internally with the fall of the start bit, which it on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse timing is shown in figure 14.24.

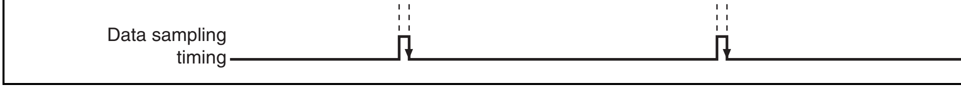


**Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode**

The receive margin in asynchronous mode can therefore be expressed as shown in ec

**Equation 1:**

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1+F) \right| \times 100 \%$$

Where: M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

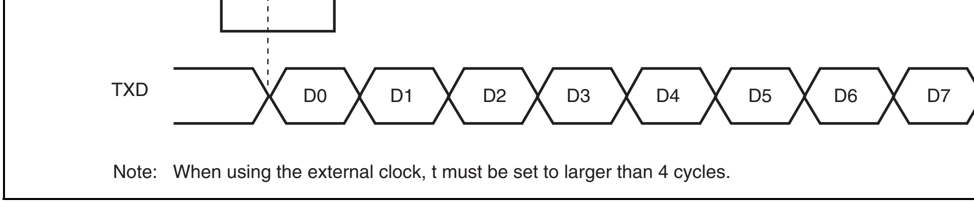
From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by ec

**Equation 2:**

When D = 0.5 and F = 0:

$$\begin{aligned} M &= (0.5 - 1/(2 \times 16)) \times 100\% \\ &= 46.875\% \end{aligned}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% tc



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 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 operati 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 se for SCI operation to be halted. Register access is enabled by clearing module standby mo details, refer to section 24, Power-Down Modes.

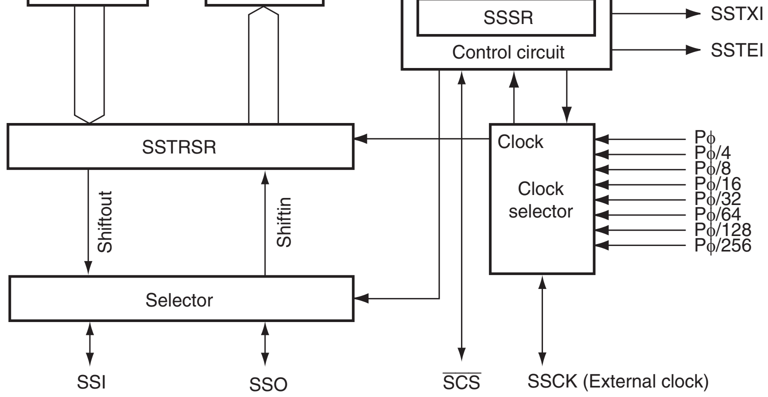
- Choice of SSU mode and clock synchronous mode
- Choice of master mode and slave mode
- Choice of standard mode and bidirectional mode
- Synchronous serial communication with devices with different clock polarity and clo
- Choice of 8/16/32-bit width of transmit/receive data
- Full-duplex communication 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 conf

  The data transfer controller (DTC) can be activated by a transmit data register empty a receive data full request to transfer data.
- Module standby mode can be set



[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**

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| SS enable register | SSER | R/W | H'00 | H'FFFFCD03 | 8 |
| SS status register | SSSR | R/W | H'04 | H'FFFFCD04 | 8, 1 |
| 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, 1 |
| 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, 1 |
| 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, 1 |
| SS receive data register 1 | SSRDR1 | R | H'00 | H'FFFFCD0B | 8 |
| SS receive data register 2 | SSRDR2 | R | H'00 | H'FFFFCD0C | 8, 1 |
| SS receive data register 3 | SSRDR3 | R | H'00 | H'FFFFCD0D | 8 |

|  |  |  |  |  |
|---|---|---|---|---|
| 7 | MSS | 0 | R/W | Master/Slave Device Select<br>Selects that this module is used in master mo<br>slave mode. When master mode is selected,<br>clocks are output from the SSCK pin. When th<br>in SSSR is set, this bit is automatically cleare<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 out<br>used or one of them is used. However, transn<br>and reception are not performed simultaneou<br>bidirectional mode is selected. For details, se<br>15.4.3, Relationship between Data Input/Outp<br>and Shift Register.<br>0: Standard mode (two pins are used for data<br>output)<br>1: Bidirectional mode (one pin is used for data<br>output) |
| 5 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s<br>always be 0. |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | 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,<br>SOL bit to 1 or clear the SOL bit to 0 after clea<br>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<br>bit is always read as 1. |
| 2 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value sh<br>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 $\overline{\text{SCS}}$ inpu<br>output.<br>00: Setting prohibited<br>01: Setting prohibited<br>10: Function as $\overline{\text{SCS}}$ automatic input/output (fu<br>$\overline{\text{SCS}}$ input before and after transfer and ou<br>low level during transfer)<br>11: Function as $\overline{\text{SCS}}$ automatic output (outputs<br>level before and after transfer and outputs<br>level during transfer) |

| | | | | |
|---|---|---|---|---|
| 7 | FCLRM | 0 | R/W | Flag Clear Mode<br>Selects whether the SSRXI and SSTXI interru are cleared on writing to SSTDR or reading fr SSRDR or on completion of DTC transfer. Wh the DTC, set this bit to 0.<br>0: Flags are cleared when DTC transfer is co (except when transfer counter value is H'0<br>1: Flags are cleared on SSTDR or SSRDR ac |
| 6 | SSUMS | 0 | R/W | Selects transfer mode from SSU mode and cl 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 int sequencer. After that, this bit is automatically The ORER, TEND, TDRE, RDRF, and CE bit and the TE and RE bits in SSER are also initi Values of other bits for SSU registers are held<br>To stop transfer, set this bit to 1 to reset the S internal sequencer. |
| 4 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va 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 |

|  |  |  |  |  |
|---|---|---|---|---|
| 7 | MLS | 0 | R/W | MSB First/LSB First Select<br>Selects that the serial data is transmitted in MS 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 a mode<br>1: Low output in idle mode, and high output in a 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 valu always be 0. |

101: Pϕ/64

110: Pϕ/128

111: Pϕ/256

| | | | | |
|---|---|---|---|---|
| 7 | TE | 0 | R/W | Transmit Enable<br>When this bit is set to 1, transmission is enable |
| 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 valu always be 0. |
| 3 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br>When this bit is set to 1, a SSTEI interrupt requ enabled. |
| 2 | TIE | 0 | R/W | Transmit Interrupt Enable<br>When this bit is set to 1, a SSTXI interrupt requ enabled. |
| 1 | RIE | 0 | R/W | Receive Interrupt Enable<br>When this bit is set to 1, an SSRXI interrupt rec 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 requ enabled. |

| | | | | |
|---|---|---|---|---|
| 7 | | 0 | R | Reserved<br>This bit is always read as 0. The write value s always be 0. |
| 6 | ORER | 0 | R/W | Overrun Error<br>If the next data is received while RDRF = 1, a error occurs, indicating abnormal termination. stores 1-frame receive data before an overrun occurs and loses data to be received later. W = 1, consecutive serial reception cannot be co Serial transmission cannot be continued, eith<br>[Setting condition]<br>• When one byte of the next reception is co 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 va always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | [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 trans<br>[Setting conditions]<br>• When the TE bit in SSER is 0<br>• When data is transferred from SSTDR to S 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 in and transmit data is written to SSTDR whil DISEL bit in MRB of the DTC is 0 (except w DTC transfer counter value is H'0000) |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | and receive data is read into SSRDR whil<br>DISEL bit in MRB of the DTC is 0 (except<br>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 $\overline{\text{SCS}}$ pin with = 0 (SSU mode) and MSS = 1 (master mode)<br>If the $\overline{\text{SCS}}$ pin level changes to 1 with SSUMS mode) and MSS = 0 (slave mode), an incomp occurs because it is determined that a master has terminated the transfer. In addition, when 0 (SSU mode) and MSS = 0 (slave mode) and serial receive operation starts while RDRF = incomplete error occurs even if the data recei SSRDR is read before the completion of rece RDRF is cleared to 0 before the $\overline{\text{SCS}}$ pin is se Data reception does not continue while the C to 1. Serial transmission also does not contin the SSU internal sequencer by setting the SR SSCRL to 1 before resuming transfer after inc error.<br>[Setting conditions]<br>• When a low level is input to the $\overline{\text{SCS}}$ pin i mode (the MSS bit in SSCRH is set to 1)<br>• When the $\overline{\text{SCS}}$ pin is changed to 1 during slave mode (the MSS bit in SSCRH is cle<br>• When in slave mode (MSS = 0 in SSCRH serial receive operation starts while RDRF data is read from SSRDR before the comp reception, after which the $\overline{\text{SCS}}$ pin is set t<br>[Clearing condition]<br>• When writing 0 after reading CE = 1 |

| | | | | |
|---|---|---|---|---|
| 7 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu<br>always be 0. |
| 4 | TENDSTS | 0 | R/W | Selects the timing of setting the TEND bit (valid<br>and master mode).<br>0: Sets the TEND bit when the last bit is being<br>transmitted<br>1: Sets the TEND bit after the last bit is transm |
| 3 | SCSATS | 0 | R/W | Selects the assertion timing of the $\overline{\text{SCS}}$ pin (va<br>SSU and master mode).<br>0: Min. values of $t_{LEAD}$ and $t_{LAG}$ are 1/2 × $t_{SUcyc}$<br>1: Min. values of $t_{LEAD}$ and $t_{LAG}$ are 3/2 × $t_{SUcyc}$ |
| 2 | SSODTS | 0 | R/W | Selects the data output timing of the SSO pin (<br>SSU and master mode)<br>0: While BIDE = 0, MSS = 1, and TE = 1 or wh<br>= 1, TE = 1, and RE = 0, the SSO pin output<br>1: While BIDE = 0, MSS = 1, and TE = 1 or wh<br>= 1, TE = 1, and RE = 0, the SSO pin output<br>while the $\overline{\text{SCS}}$ pin is driven low |
| 1, 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu<br>always be 0. |

Although SSTDR can always be read from or written to by the CPU and DTC, to achiev serial transmission, write transmit data to SSTDR after confirming that the TDRE bit in 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 (Invali |
| SSTDR0 | Valid | Valid | Valid | Invalid |
| SSTDR1 | Invalid | Valid | Valid | Invalid |
| SSTDR2 | Invalid | Invalid | Valid | Invalid |
| SSTDR3 | Invalid | Invalid | Valid | Invalid |

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** |
| SSRDR0 | Valid | Valid | Valid | Invalid |
| SSRDR1 | Invalid | Valid | Valid | Invalid |
| SSRDR2 | Invalid | Invalid | Valid | Invalid |
| SSRDR3 | Invalid | Invalid | Valid | Invalid |

SSRDR. SSTRSR cannot be directly accessed by the CPU.

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

The relationship of clock phase, polarity, and transfer data depends on the combination o CPOS and CPHS bits in SSMR when the value of the SSUMS bit in SSCRL is 0. Figure shows the relationship. When SSUMS = 1, the CPHS setting is invalid although the CPO is valid.

Setting the MLS bit in SSMR selects that MSB or LSB first communication. When MLS is transferred from the LSB to the MSB. When MLS = 1, data is transferred from the MS LSB.

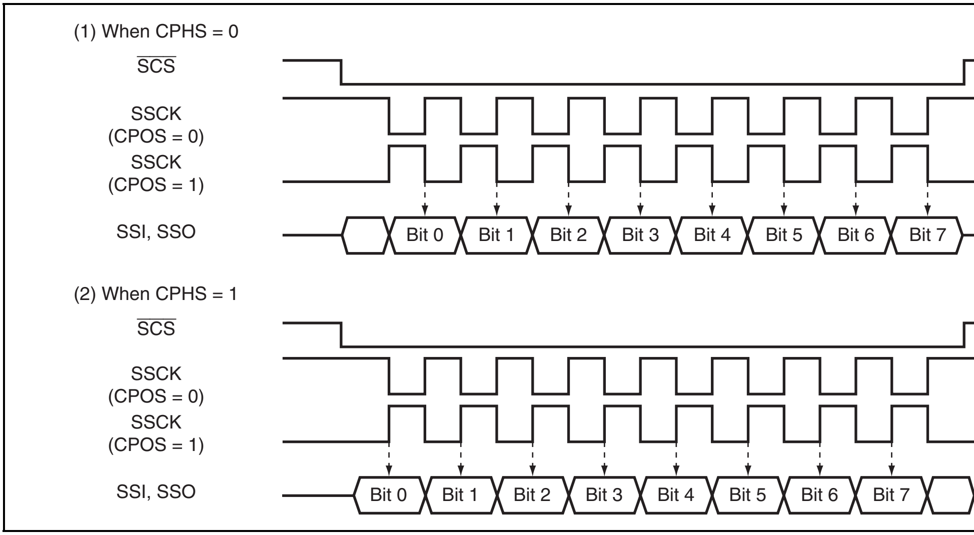


**Figure 15.2 Relationship of Clock Phase, Polarity, and Data**

The SSU transmits and receives serial data from the SSO pin regardless of master or sla
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 no
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 pi
operating with SSUMS = 1. The SSCK pin outputs the internal clock when MSS = 1 and
as an input pin when MSS = 0 (see figures 15.3 (5) and (6)).

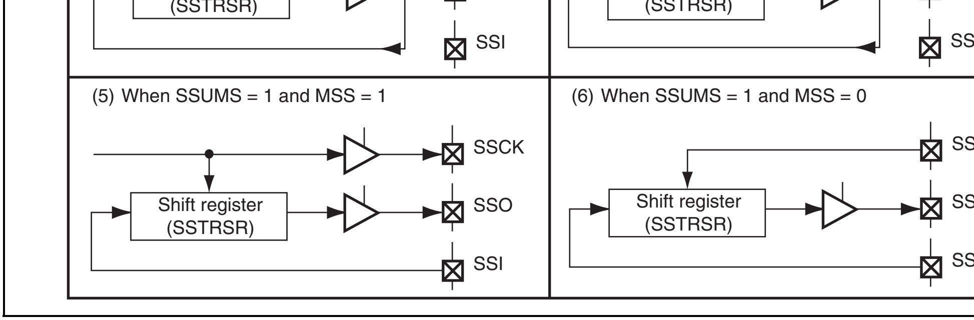


**Figure 15.3 Relationship between Data Input/Output Pins and the Shift Regis**

| | | | | | | |
|---|---|---|---|---|---|---|
| SSU communication mode | 0 | 0 | 0 | 0 | 1 | — |
| | | | | 1 | 0 | Output |
| | | | | | 1 | Output |
| | | | 1 | 0 | 1 | Input |
| | | | | 1 | 0 | — |
| | | | | | 1 | Input |
| SSU (bidirectional) communication mode | 0 | 1 | 0 | 0 | 1 | — |
| | | | | 1 | 0 | — |
| | | | 1 | 0 | 1 | — |
| | | | | 1 | 0 | — |
| Clock synchronous communication mode | 1 | 0 | 0 | 0 | 1 | Input |
| | | | | 1 | 0 | — |
| | | | | | 1 | Input |
| | | | 1 | 0 | 1 | Input |
| | | | | 1 | 0 | — |
| | | | | | 1 | Input |

[Legend]

—: Not used as SSU pin

**Table 15.7 Communication Modes and Pin States of $\overline{\text{SCS}}$ Pin**

| Communication Mode | Register Setting | | | | Pin |
|---|---|---|---|---|---|
| | SSUMS | MSS | CSS1 | CSS0 | $\overline{\text{SCS}}$ |
| SSU communication mode | 0 | 0 | x | x | Inpu |
| | | 1 | 0 | 0 | — |
| | | | 0 | 1 | — |
| | | | 1 | 0 | Aut<br>inpu |
| | | | 1 | 1 | Out |
| Clock synchronous communication mode | 1 | x | x | x | — |

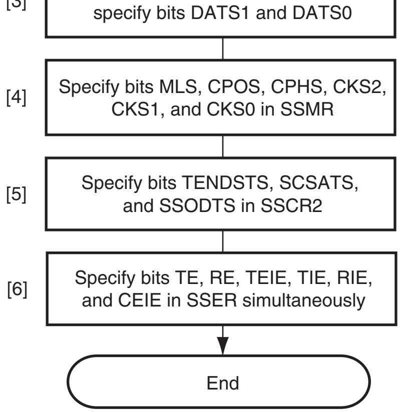
[Legend]

x: Don't care

—: Not used as SSU pin

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 T bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the R does not change the values of the RDRF and ORER bits and SSRDR. Those bits previous values.


[3]
specify bits DATS1 and DATS0
[4]
Specify bits MLS, CPOS, CPHS, CKS2, CKS1, and CKS0 in SSMR
[5]
Specify bits TENDSTS, SCSATS, and SSODTS in SSCR2
[6]
Specify bits TE, RE, TEIE, TIE, RIE, and CEIE in SSER simultaneously
End

output on the SSO pin.

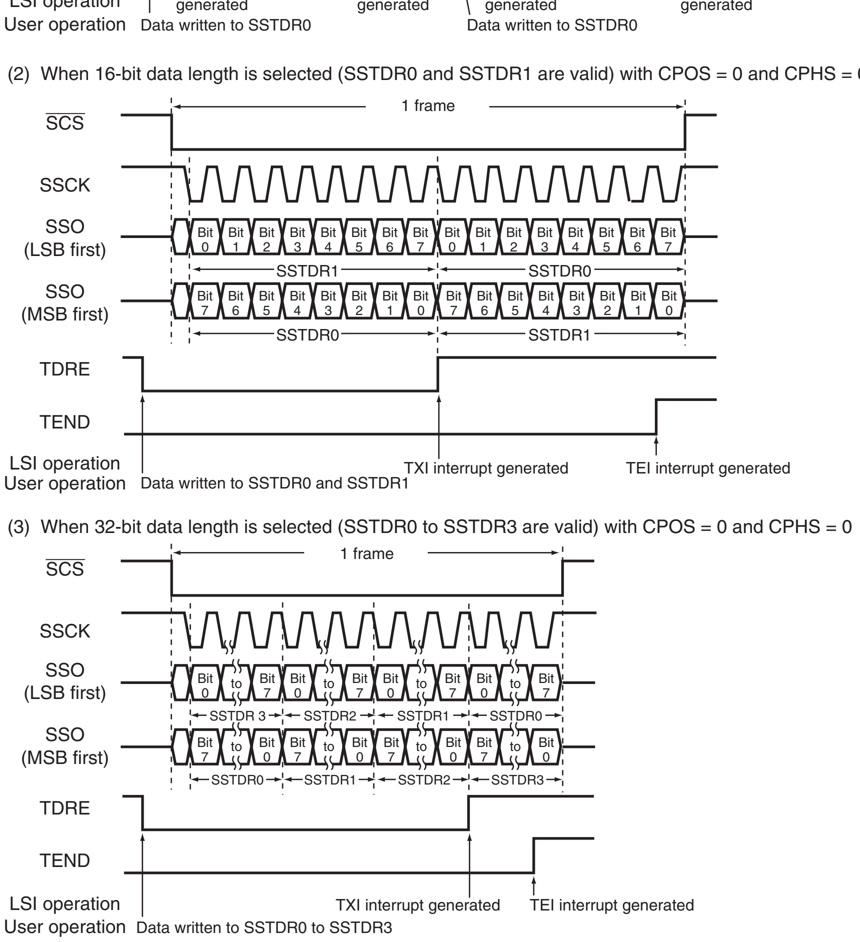
[6] Enables/disables interrupt requests to the CPU.

**Figure 15.4 Example of Initial Settings in SSU Mode**

the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit t
starts transmission. At this time, if the TIE bit in SSER is set to 1, a TXI interrupt is gen

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transfe
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. A
if the TEIE bit is set to 1, a TEI interrupt is generated. After transmission, the output lev
SSCK pin is fixed high when CPOS = 0 and low when CPOS = 1.

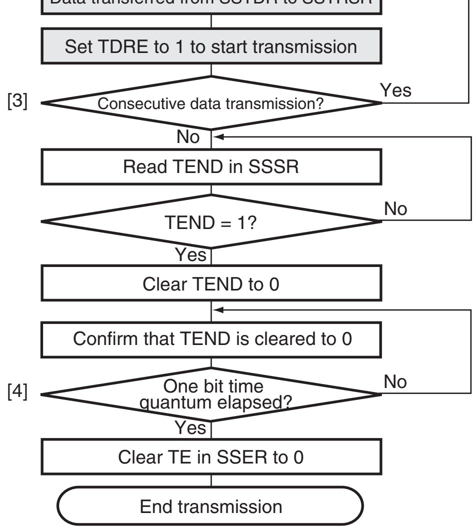
While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the O
is cleared to 0 before transmission.



(2) When 16-bit data length is selected (SSTDR0 and SSTDR1 are valid) with CPOS = 0 and CPHS = 0

(3) When 32-bit data length is selected (SSTDR0 to SSTDR3 are valid) with CPOS = 0 and CPHS = 0

**Figure 15.5 Example of Transmission Operation (SSU Mode)**


Data transferred from SSTDR to SSTRSR
Set TDRE to 1 to start transmission
[3]
Consecutive data transmission?
Yes
No
Read TEND in SSSR
TEND = 1?
No
Yes
Clear TEND to 0
Confirm that TEND is cleared to 0
[4]
One bit time quantum elapsed?
No
Yes
Clear TE in SSER to 0
End transmission

To end data transmission, confirm that the TEND b
to 0. After completion of transmitting the last bit, cl
bit to 0.

Note: Hatching boxes represent SSU internal op

**Figure 15.6 Flowchart Example of Data Transmission (SSU Mode)**

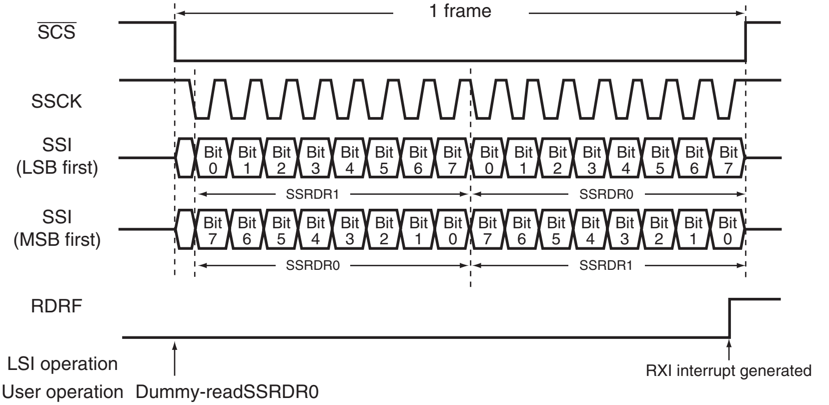
stored in SSRDR. At this time, if the RIE bit in SSER is set to 1, an RXI interrupt is gene
The RDRF bit is automatically cleared to 0 by reading SSRDR.

When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORE
SSSR is set to 1. This indicates that an overrun error (OEI) has occurred. At this time, dat
reception is stopped. While the ORER bit in SSSR is set to 1, reception is not performed.
resume the reception, clear the ORER bit to 0.

When setting the SSU to slave mode to perform continuous reception, read SSRDR befor
the next receive operation. If the next receive operation starts before SSRDR is read and
cleared to 0, and SSRDR is read before reception completes, CE in SSSR is set to 1 after
completion of reception.

In addition, if the next receive operation starts before SSRDR is read and RDRF is cleare
and SSRDR is not read until after reception completes, the receive data is discarded even
neither CE nor ORER in SSSR is set to 1.

(2) When 16-bit data length is selected (SSRDR0 and SSRDR1 are valid) with CPOS = 0 and CPHS


1 frame
SCS
SSCK
SSI
(LSB first)
Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7 Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7
SSRDR1
SSRDR0
SSI
(MSB first)
Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0
SSRDR0
SSRDR1
RDRF
LSI operation
User operation
Dummy-readSSRDR0
RXI interrupt generated

(3) When 32-bit data length is selected (SSRDR0 to SSRDR3 are valid) with CPOS = 0 and CPHS =

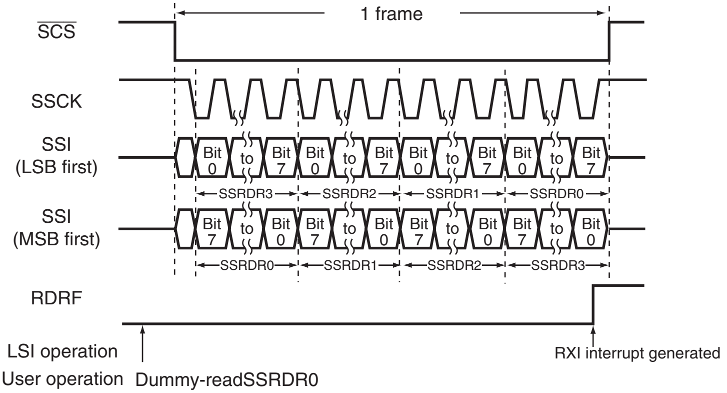
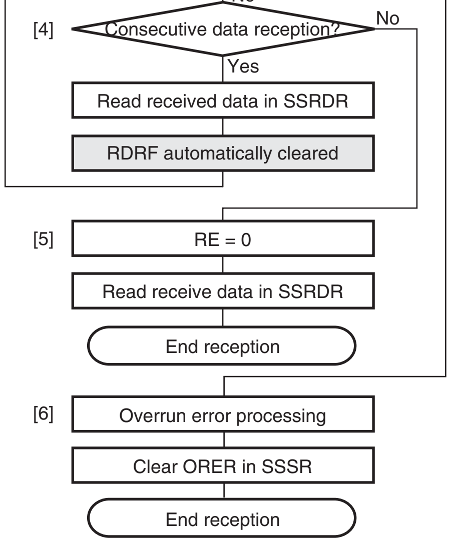


**Figure 15.7 Example of Reception Operation (SSU Mode)**


No
[4]
Consecutive data reception?
No
Yes
Read received data in SSRDR
RDRF automatically cleared
[5]
RE = 0
Read receive data in SSRDR
End reception
[6]
Overrun error processing
Clear ORER in SSSR
End reception

while the RDRF flag is set to 1 and then read rec in SSRDR.
The next single reception starts after reading rec in SSRDR.

[5] To complete reception:
To complete reception, read receive data after cl RE bit to 0. When reading SSRDR without cleari bit, reception is resumed.

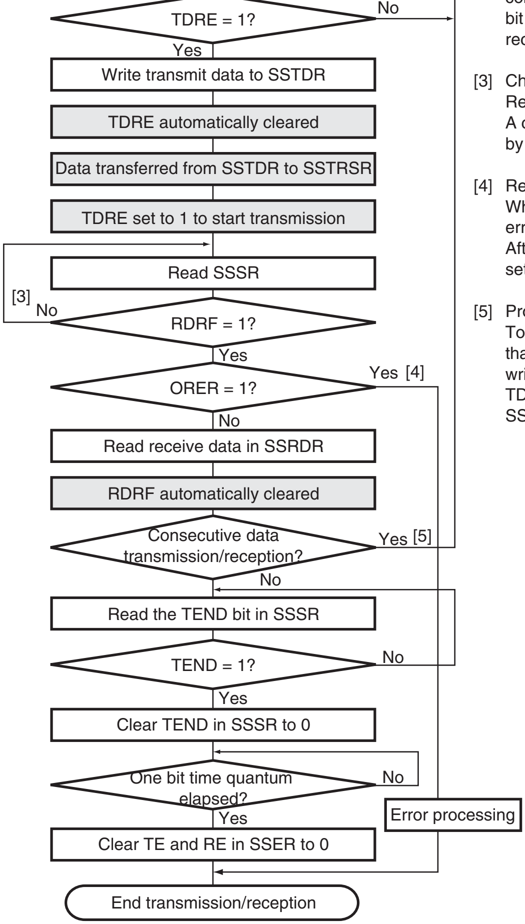
Note: Hatching boxes represent SSU internal operations.

**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 a mentioned above. The data transmission/reception is started by writing transmit data to S 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


No
TDRE = 1?
Yes
Write transmit data to SSTDR
TDRE automatically cleared
Data transferred from SSTDR to SSTRSR
TDRE set to 1 to start transmission
Read SSSR
[3]
No
RDRF = 1?
Yes
Yes
[4]
ORER = 1?
No
Read receive data in SSRDR
RDRF automatically cleared
Consecutive data transmission/reception?
Yes
[5]
No
Read the TEND bit in SSSR
TEND = 1?
No
Yes
Clear TEND in SSSR to 0
One bit time quantum elapsed?
No
Yes
Error processing
Clear TE and RE in SSER to 0
End transmission/reception

confirming that the TDRE bit in SSSR is 1. The TD 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 noti by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designat error processing after reading the ORER bit in SSS After that, clear the ORER bit to 0. While the OREF set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/recep
To continue serial data transmission/reception, cor that the TDRE bit is 1 meaning that SSTDR is reac written to. After that, data can be written to SSTDR TDRE bit is automatically cleared to 0 by writing da SSTDR.

Note: Hatching boxes represent SSU internal operations.

**Figure 15.9 Flowchart Example of Simultaneous Transmission/Reception (SSU**

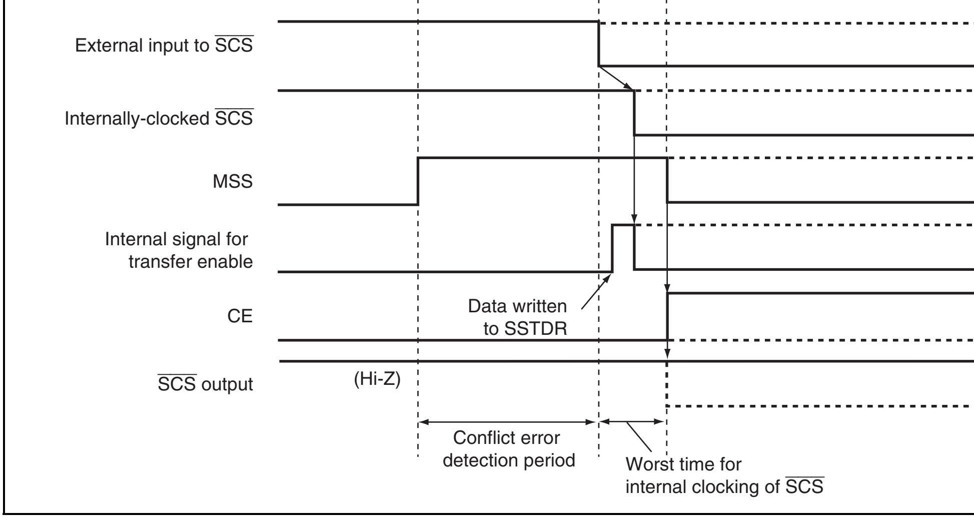
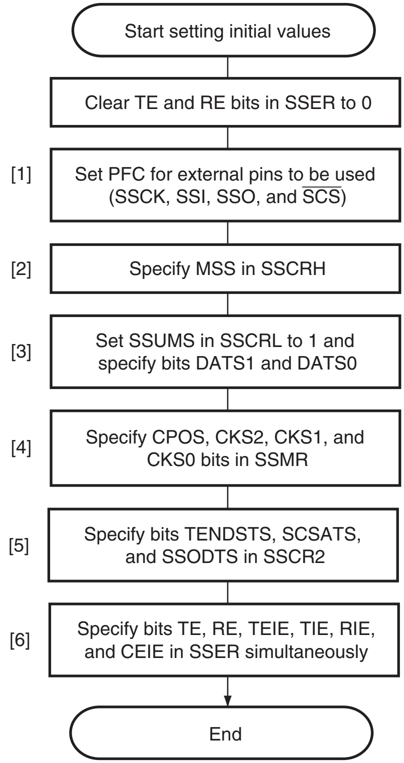

External input to $\overline{\text{SCS}}$
Internally-clocked $\overline{\text{SCS}}$
MSS
Internal signal for transfer enable
CE
Data written to SSTDR
$\overline{\text{SCS}}$ output
(Hi-Z)
Conflict error detection period
Worst time for internal clocking of $\overline{\text{SCS}}$

**Figure 15.10 Conflict Error Detection Timing (Before Transfer)**

Figure 15.11 Conflict Error Detection Timing (After Transfer End)

does not change the values of the RDRF and ORER bits and SSRDR. Those bits r previous values.


Start setting initial values
Clear TE and RE bits in SSER to 0
[1]
Set PFC for external pins to be used (SSCK, SSI, SSO, and $\overline{SCS}$)
[2]
Specify MSS in SSCRH
[3]
Set SSUMS in SSCRL to 1 and specify bits DATS1 and DATS0
[4]
Specify CPOS, CKS2, CKS1, and CKS0 bits in SSMR
[5]
Specify bits TENDSTS, SCSATS, and SSODTS in SSCR2
[6]
Specify bits TE, RE, TEIE, TIE, RIE, and CEIE in SSER simultaneously
End

[1] Make appropriate settings in the PFC for the external pins to be used.

[2] Specify master/slave mode selection and SSCK pin selection.

[3] Selects clock synchronous communication mode and specify transmit/receive data length.

[4] Specify clock polarity selection and transfer clock rate selection.

[5] Specify timing of TEND bit setting, $\overline{SCS}$ pin assertion, a output on the SSO pin.

[6] Enables/disables interrupt requests to the CPU.

**Figure 15.12 Example of Initial Settings in Clock Synchronous Communication**

starts transmission. At this time, if the TIE bit in SSER is set to 1, a TXI interrupt is gen

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transf 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. A 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 C is cleared to 0 before transmission.

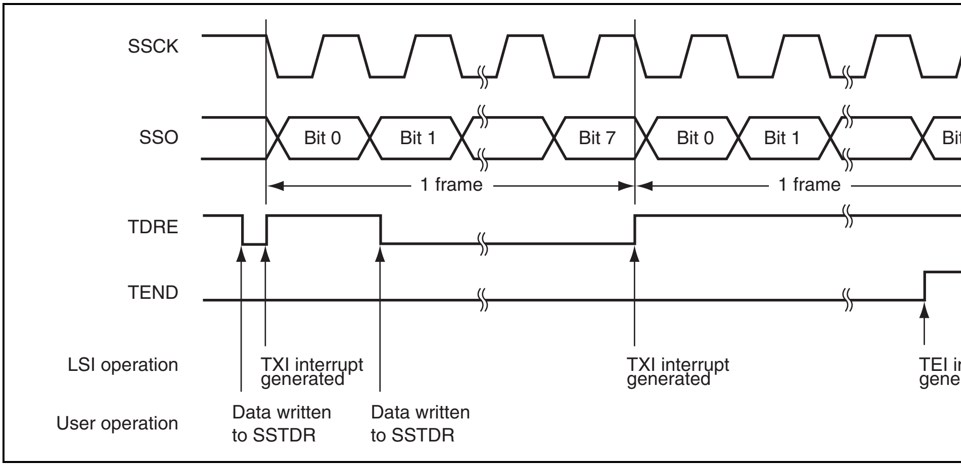
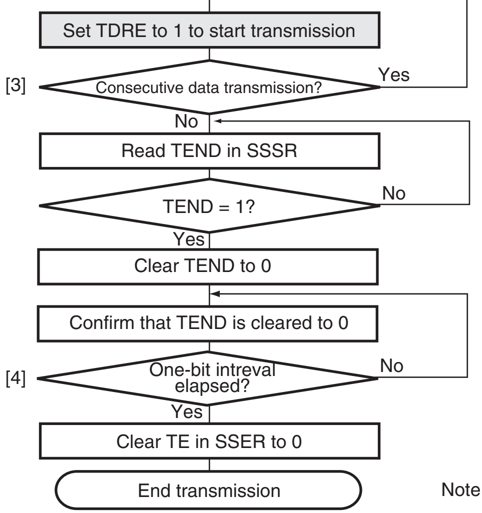


**Figure 15.13 Example of Transmission Operation (Clock Synchronous Communication Mode)**


Set TDRE to 1 to start transmission
[3]
Consecutive data transmission?
Yes
No
Read TEND in SSSR
TEND = 1?
No
Yes
Clear TEND to 0
Confirm that TEND is cleared to 0
[4]
One-bit intreval elapsed?
No
Yes
Clear TE in SSER to 0
End transmission

to 0. After completion of transmitting the last bit, clea
bit to 0.

Note: Hatched boxes represent SSU internal operations.

**Figure 15.14 Flowchart Example of Transmission Operation (Clock Synchronous Communication Mode)**

bit is automatically cleared to 0 by reading SSRDR.

When setting the SSU to slave mode to perform continuous reception, read SSRDR befo the next receive operation. If the next receive operation starts before SSRDR is read and cleared to 0, the integrity of subsequent data cannot be guaranteed.

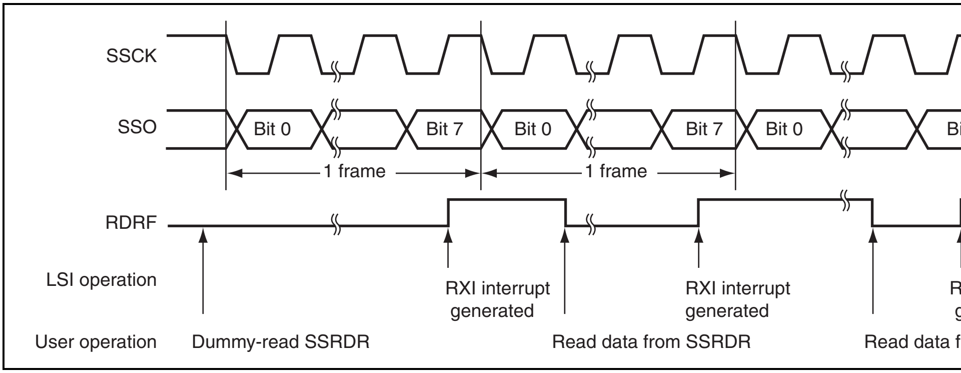
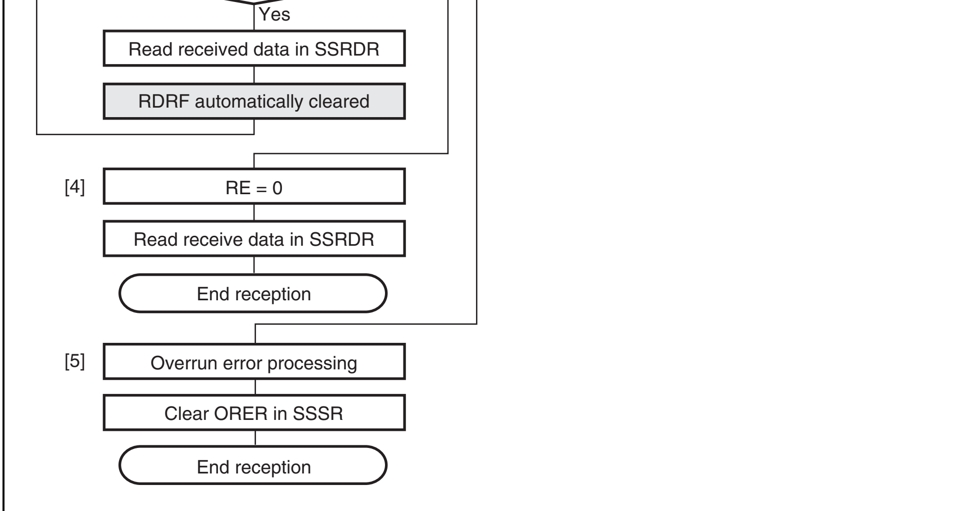


**Figure 15.15 Example of Reception Operation (Clock Synchronous Communication Mode)**



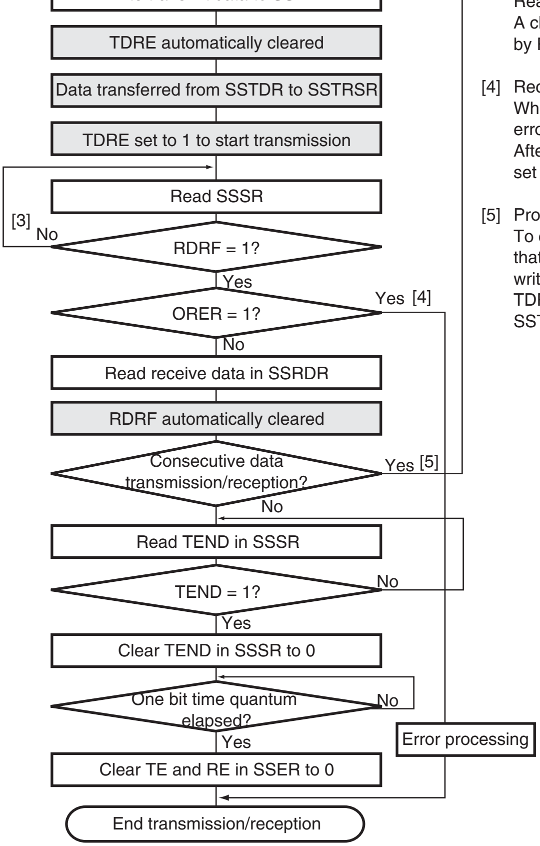
Note: Hatching boxes represent SSU internal operations.

**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 dat transmission/reception is performed combining the data transmission and data reception a mentioned above. The data transmission/reception is started by writing transmit data to S 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 transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting th RE bits to 1.


TDRE automatically cleared
Data transferred from SSTDR to SSTRSR
TDRE set to 1 to start transmission
Read SSSR
[3]
No
RDRF = 1?
Yes
ORER = 1?
Yes [4]
No
Read receive data in SSRDR
RDRF automatically cleared
Consecutive data transmission/reception?
Yes [5]
No
Read TEND in SSSR
TEND = 1?
No
Yes
Clear TEND in SSSR to 0
One bit time quantum elapsed?
No
Yes
Error processing
Clear TE and RE in SSER to 0
End transmission/reception

Read SSSR confirming that the RDRF bit is 1.
A change of the RDRF bit (from 0 to 1) can be not
by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designa
error processing after reading the ORER bit in SS
After that, clear the ORER bit to 0. While the ORE
set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/rece
To continue serial data transmission/reception, co
that the TDRE bit is 1 meaning that SSTDR is rea
written to. After that, data can be written to SSTDF
TDRE bit is automatically cleared to 0 by writing d
SSTDR.

Note: Hatching boxes represent SSU internal operations.

**Figure 15.17 Flowchart Example of Simultaneous Transmission/Receptio (Clock Synchronous Communication Mode)**

When an interrupt condition shown in table 15.8 is satisfied, an interrupt is requested. Cle interrupt source by CPU or DTC data transfer.

**Table 15.8 SSU Interrupt Sources**

| Abbreviation | Interrupt Source | Symbol | Interrupt Condition | DTC Ac |
|---|---|---|---|---|
| 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) | — |

SSCRL register. If accessed, transmission or reception thereafter may not be performed

### 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 every frame. If the $\overline{\text{SCS}}$ pin is kept asserted (low level) for more than one frame, transm 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 S data register (SSRDR) before each next round of reception starts (i.e. before an external 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) been set to 1 but before the SSRDR has been read, and the SSRDR is read before the rec one frame is complete, the conflict/incomplete error bit in SSSR will be set to 1 on comp reception.

Furthermore, when the next round of reception starts after the receive-data full (RDRF) been set to 1 and before the SSRDR has been read, and the SSRDR has not been read by of the reception of the frame, the CE and overflow-error (ORER) bits will not have been the received data will be discarded.

Further note that this point for caution does not apply to simultaneous transmission and in SSU slave mode or to clock-synchronous mode.

### 15.6.6 Note on DTC Transfers

When a DTC transfer occurs with SSTXI as the activation source, TDRE is not cleared w 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 o DTC's transfer counter is set to H'0001 and DISEL is set to 1. In this case, clearing the fl interrupt handler may cause the SSU to start communication operation a second time.

- Selection of $I^2C$ format or clock synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independe each other, the continuous transmission/reception can be performed.
- Module standby mode can be set

**$I^2C$ bus format:**

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronize automatically.

  If transmission/reception is not yet possible, set the SCL to low until preparations ar completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full slave-address match), arbitration lost, NACK detection, and stop condition detection

  The data transfer controller (DTC) can be activated by a transmit-data-empty reques 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 function is selected.

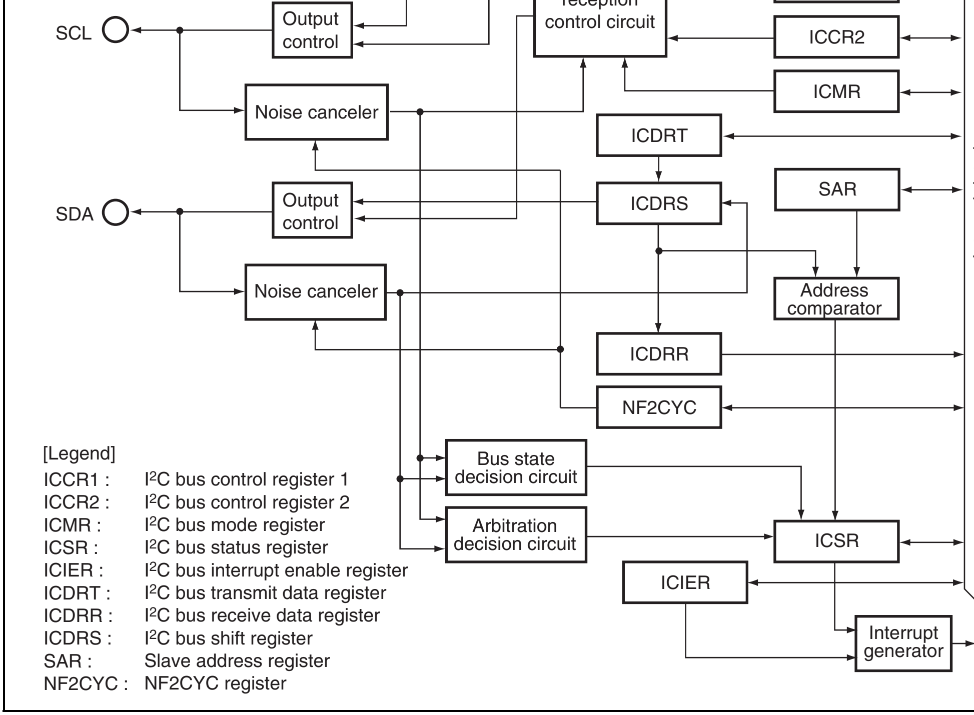

reception
control circuit
SCL
Output
control
ICCR2
ICMR
Noise canceler
ICDRT
SDA
Output
control
ICDRS
SAR
Noise canceler
Address
comparator
ICDRR
NF2CYC
Bus state
decision circuit
Arbitration
decision circuit
ICSR
ICIER
Interrupt
generator
[Legend]
ICCR1 : I$^2$C bus control register 1
ICCR2 : I$^2$C bus control register 2
ICMR : I$^2$C bus mode register
ICSR : I$^2$C bus status register
ICIER : I$^2$C bus interrupt enable register
ICDRT : I$^2$C bus transmit data register
ICDRR : I$^2$C bus receive data register
ICDRS : I$^2$C bus shift register
SAR : Slave address register
NF2CYC : NF2CYC register

**Figure 16.1 Block Diagram of I$^2$C Bus Interface 2**

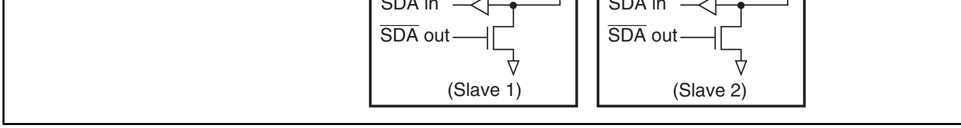


**Figure 16.2 External Circuit Connections of I/O Pins**

| | | | | | |
|---|---|---|---|---|---|
| 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 controls transmission or reception, and selects master or slave mode, transmission or rec 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. SDA pins are bus drive state.) |

| | | | | |
|---|---|---|---|---|
| 4 | TRS | 0 | R/W | Transmit/Receive Select<br>In master mode with the I$^2$C bus format, when arbitration is lost, MST and TRS are both rese hardware, causing a transition to slave receiv Modification of the TRS bit should be made be transfer frames.<br>When seven bits after the start condition is iss slave receive mode match the slave address SAR and the 8th bit is set to 1, TRS is automa set to 1. If an overrun error occurs in master r mode with the clock synchronous serial forma cleared and the mode changes to slave recei<br>Operating modes are described below accord MST and TRS combination. When clock sync serial format is selected and MST = 1, clock i<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 nec transfer rate (table 16.3) in master mode. In s mode, these bits should be used to specify th setup time in transmission mode. The setup t to 10 tpcyc when CKS3 = 0 or 20 tpcyc when 1 (tpcyc is one Pϕ cycle). |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 0 | Pϕ/112 | 89.3 kHz | 143 kHz | 179 kHz | 223 kHz | 295 kHz |
| | | | 1 | Pϕ/128 | 78.1 kHz | 125 kHz | 156 kHz | 195 kHz | 258 kHz |
| 1 | 0 | 0 | 0 | Pϕ/112 | 89.3 kHz | 143 kHz | 179 kHz | 223 kHz | 295 kHz |
| | | | 1 | Pϕ/160 | 62.5 kHz | 100 kHz | 125 kHz | 156 kHz | 206 kHz |
| | | 1 | 0 | Pϕ/192 | 52.1 kHz | 83.3 kHz | 104 kHz | 130 kHz | 172 kHz |
| | | | 1 | Pϕ/256 | 39.1 kHz | 62.5 kHz | 78.1 kHz | 97.7 kHz | 129 kHz |
| | 1 | 0 | 0 | Pϕ/320 | 31.3 kHz | 50.0 kHz | 62.5 kHz | 78.1 kHz | 103 kHz |
| | | | 1 | Pϕ/400 | 25.0 kHz | 40.0 kHz | 50.0 kHz | 62.5 kHz | 82.5 kHz |
| | | 1 | 0 | Pϕ/448 | 22.3 kHz | 35.7 kHz | 44.6 kHz | 55.8 kHz | 73.7 kHz |
| | | | 1 | Pϕ/512 | 19.5 kHz | 31.3 kHz | 39.1 kHz | 48.8 kHz | 64.5 kHz |

|  |  |  |  |  |
|---|---|---|---|---|
| 7 | BBSY | 0 | R/W | Bus Busy<br>This bit enables to confirm whether the I$^2$C bus occupied or released and to issue start/stop co in master mode. With the clock synchronous se format, this bit is always read as 0. With the I$^2$C format, this bit is set to 1 when the SDA level c from high to low under the condition of SCL = h assuming that the start condition has been issu bit is cleared to 0 when the SDA level changes to high under the condition of SCL = high, assu that the stop condition has been issued. To iss condition, simultaneously write 1 to BBSY and SCP. Follow this procedure also when transmit repeated start condition. To issue a stop condit 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 con master mode.<br>To issue a start condition, simultaneously write BBSY and 0 to SCP. A repeated start condition issued in the same way. To issue a stop condit simultaneously write 0 to BBSY and 0 to SCP. is always read as 1. Even if 1 is written to this b data will not be stored. |

| | | | | |
|---|---|---|---|---|
| 4 | SDAOP | 1 | R/W | SDAO Write Protect<br>This bit controls change of output level of the by modifying the SDAO bit. To change the ou clear SDAO and SDAOP to 0 or set SDAO to clear SDAOP to 0. This bit is always read as |
| 3 | SCLO | 1 | R | This bit monitors SCL output level. When SCL SCL pin outputs high. When SCLO is 0, SCL outputs low. |
| 2 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value s always be 1. |
| 1 | IICRST | 0 | R/W | IIC Control Part Reset<br>This bit resets the control part except for I$^2$C r this bit is set to 1 when hang-up occurs becau communication failure during I$^2$C operation, sc registers and control part can be reset. |
| 0 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value s always be 1. |

| | | | | |
|---|---|---|---|---|
| 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 I$^2$C bus format is used |
| 6 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value sh always be 0. |
| 5, 4 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write valu always be 1. |
| 3 | BCWP | 1 | R/W | BC Write Protect<br>This bit controls the BC2 to BC0 modifications. modifying BC2 to BC0, this bit should be cleare In clock synchronous serial mode, BC should r modified.<br>0: When writing, values of BC2 to BC0 are set.<br>1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are inv |

condition is detected. These bits are cleared b power-on reset and in standby mode. These b also cleared by setting IICRST of ICCR2 to 1. clock synchronous serial format, these bits sh be modified.

| $I^2C$ Bus Format | Clock Synchronous Seri |
|---|---|
| 000: 9 bits | 000: 8 bits |
| 001: 2 bits | 001: 1 bit |
| 010: 3 bits | 010: 2 bits |
| 011: 4 bits | 011: 3 bits |
| 100: 5 bits | 100: 4 bits |
| 101: 6 bits | 101: 5 bits |
| 110: 7 bits | 110: 6 bits |
| 111: 8 bits | 111: 7 bits |

| 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, thi enables or disables the transmit data empty int (IITXI).<br>0: Transmit data empty interrupt request (IITXI disabled.<br>1: Transmit data empty interrupt request (IITXI enabled. |
| 6 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br>This bit enables or disables the transmit end in (IITEI) at the rising of the ninth clock while the in ICSR is 1. IITEI can be canceled by clearing TEND bit or the TEIE bit to 0.<br>0: Transmit end interrupt request (IITEI) is disa<br>1: Transmit end interrupt request (IITEI) is ena |
| 5 | RIE | 0 | R/W | Receive Interrupt Enable<br>RIE enables or disables the receive data full in request (IIRXI) when receive data is transferre ICDRS to ICDRR and the RDRF bit in ICSR is IIRXI can be canceled by clearing the RDRF o to 0.<br>0: Receive data full interrupt request (IIRXI) are disabled.<br>1: Receive data full interrupt request (IIRXI) are enabled. |

| | | | | |
|---|---|---|---|---|
| | | | | This bit enables or disables the stop conditior interrupt request (IISTPI) when the STOP bit set.<br>0: Stop condition detection interrupt request ( disabled.<br>1: Stop condition detection interrupt request ( enabled. |
| 2 | ACKE | 0 | R/W | Acknowledge Bit Judgment Select<br>0: The value of the receive acknowledge bit is and continuous transfer is performed.<br>1: If the receive acknowledge bit is 1, continu transfer is halted. |
| 1 | ACKBR | 0 | R | Receive Acknowledge<br>In transmit mode, this bit stores the acknowle that are returned by the receive device. This b be modified. This bit can be canceled by setti 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 the acknowledge timing.<br>0: 0 is sent at the acknowledge timing.<br>1: 1 is sent at the acknowledge timing. |

| | | | | |
|---|---|---|---|---|
| 7 | TDRE | 0 | R/W | Transmit Data Register Empty<br>[Setting conditions]<br>• When data is transferred from ICDRT to IC ICDRT becomes empty<br>• When TRS is set<br>• When the start condition (including retransr is issued<br>• When slave mode is changed from receive transmit mode<br>[Clearing conditions]<br>• When 0 is written to TDRE after reading TD<br>• When data is written to ICDRT<br>• DTC is activated by IITXI interrupt and the 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 format while the TDRE flag is 1<br>• When the final bit of transmit frame is sent clock synchronous serial format<br>[Clearing conditions]<br>• When 0 is written to TEND after reading TE<br>• When data is written to ICDRT<br>• DTC is activated by IITXI interrupt and the in MRB of DTC is 0. |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | 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 th device in transmission while the ACKE bit is 1<br>[Clearing condition]<br>• When 0 is written to NACKF after reading = 1 |
| 3 | STOP | 0 | R/W | Stop Condition Detection Flag<br>[Setting conditions]<br>• In master mode, when a stop condition is after frame transfer<br>• In slave mode, when a stop condition is de after the slave address in the first byte tha following the detection of a start condition matched the address set in SAR.<br>[Clearing condition]<br>• When 0 is written to STOP after reading S |

| | | | | |
|---|---|---|---|---|
| | | | | [Setting conditions]<br>• If the internal SDA and SDA pin disagree a of SCL in master transmit mode<br>• When the SDA pin outputs high in master r 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 = 1 |
| 1 | AAS | 0 | R/W | Slave Address Recognition Flag<br>In slave receive mode, this flag is set to 1 if the frame following a start condition matches bits S SVA0 in SAR.<br>[Setting conditions]<br>• When the slave address is detected in slav mode<br>• When the general call address is detected receive mode.<br>[Clearing condition]<br>• When 0 is written to AAS after reading AAS |

Note: * When NACKF = 1 is detected, be sure to clear NACKF in the transfer end pro
Until the flag is cleared, next transmission or reception cannot be started.

### 16.3.6 I$^2$C Bus Slave Address Register (SAR)

SAR is an 8-bit readable/writable register that selects the communications format and se slave address. In slave mode with the I$^2$C bus format, if the upper seven bits of SAR mat upper seven bits of the first frame received after a start condition, this module operates a 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 differing form the addresses of other slave de connected to the I$^2$C bus. |
| 0 | FS | 0 | R/W | Format Select<br>0: I$^2$C bus format is selected<br>1: Clock synchronous serial format is selectec |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 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 I$^2$C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received,
transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDR
receive-only register, therefore the CPU cannot write to this register. ICDRR is initialized

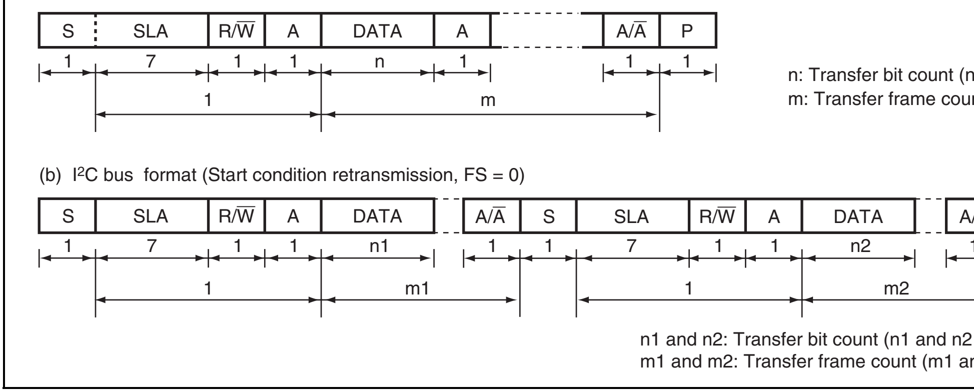
| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Initial value: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

### 16.3.9 I$^2$C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferre
ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred
ICDRS to ICDRR after data of one byte is received. This register cannot be read directly
CPU.

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

| Bit | Bit Name | Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va always be 0. |
| 0 | NF2CYC | 0 | R/W | Noise Filtering Range Select<br>0: The noise less than one cycle of the periph can be filtered out<br>1: The noise less than two cycles of the perip can be filtered out |



n: Transfer bit count (n
m: Transfer frame cour

(b) I$^2$C bus format (Start condition retransmission, FS = 0)

n1 and n2: Transfer bit count (n1 and n2
m1 and m2: Transfer frame count (m1 ar

**Figure 16.3 I$^2$C Bus Formats**

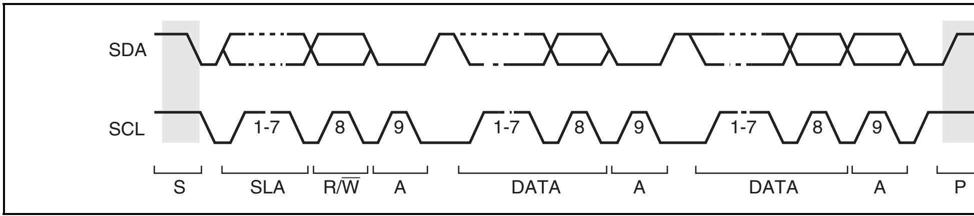


**Figure 16.4 I$^2$C Bus Timing**

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/W̄: Indicates the direction of data transfer: from the slave device to the master device 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.

issued) This generates the start condition.

3. After confirming that TDRE in ICSR has been set, write the transmit data (the first b show the slave address and $R/\overline{W}$) to ICDRT. At this time, TDRE is automatically cle 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 i at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confir slave device has been selected. Then, write second byte data to ICDRT. When ACKI the slave device has not been acknowledged, so issue the stop condition. To issue the condition, write 0 to BBSY and SCP. SCL is fixed low until the transmit data is prep 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 e byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TE NACKF.

7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mo

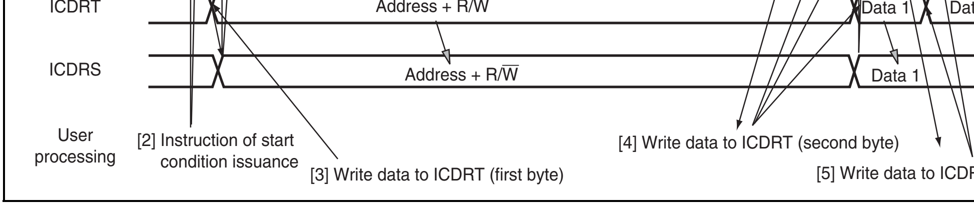

ICDRT
Address + R/W
Data 1
Dat
ICDRS
Address + R/$\overline{W}$
Data 1
User processing
[2] Instruction of start condition issuance
[3] Write data to ICDRT (first byte)
[4] Write data to ICDRT (second byte)
[5] Write data to ICDF

**Figure 16.5 Master Transmit Mode Operation Timing (1)**

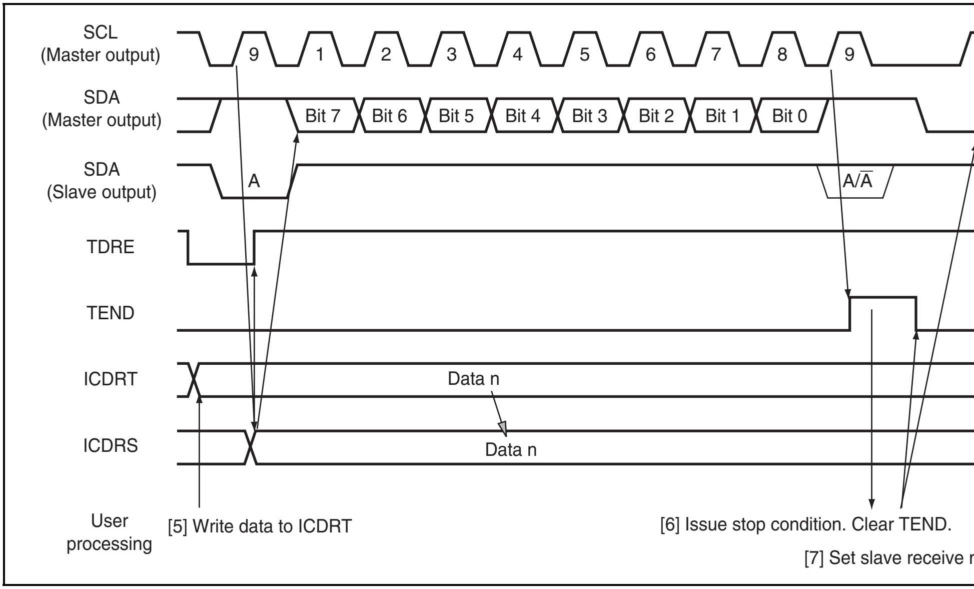


**Figure 16.6 Master Transmit Mode Operation Timing (2)**

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
   of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, a
   is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. I
   receive clock pulse falls after reading ICDRR by the other processing while RDRF i
   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
   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 c
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 I
set.

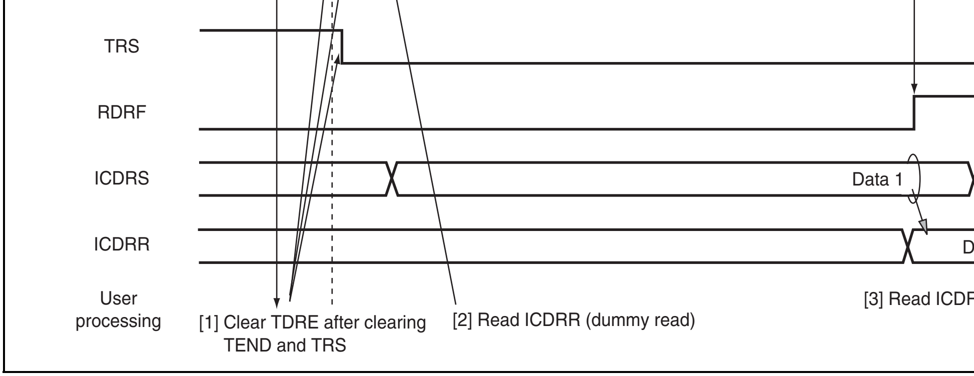

TRS
RDRF
ICDRS
Data 1
ICDRR
D
User processing
[1] Clear TDRE after clearing TEND and TRS
[2] Read ICDRR (dummy read)
[3] Read ICDF

**Figure 16.7 Master Receive Mode Operation Timing (1)**

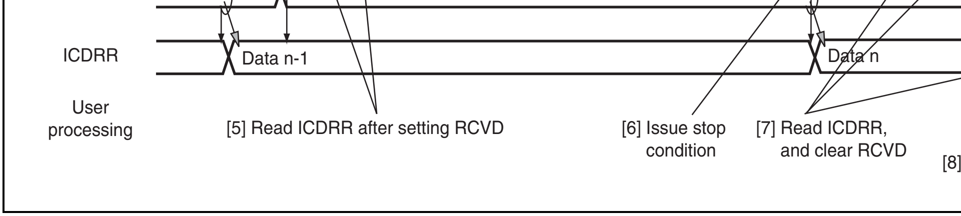

ICDRR
Data n-1
Data n
User processing
[5] Read ICDRR after setting RCVD
[6] Issue stop condition
[7] Read ICDRR, and clear RCVD
[8]

**Figure 16.8 Master Receive Mode Operation Timing (2)**

2. When the slave address matches in the first frame following detection of the start con the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise o clock pulse. At this time, if the 8th bit data (R/W) is 1, the TRS bit in ICCR1 and the 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 T set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is 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.

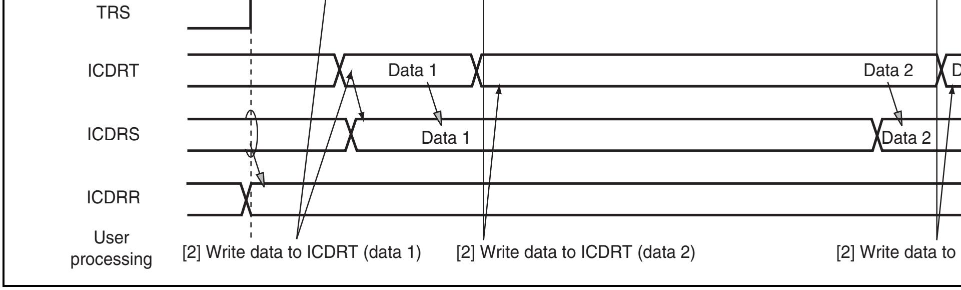


Figure 16.9 Slave Transmit Mode Operation Timing (1)

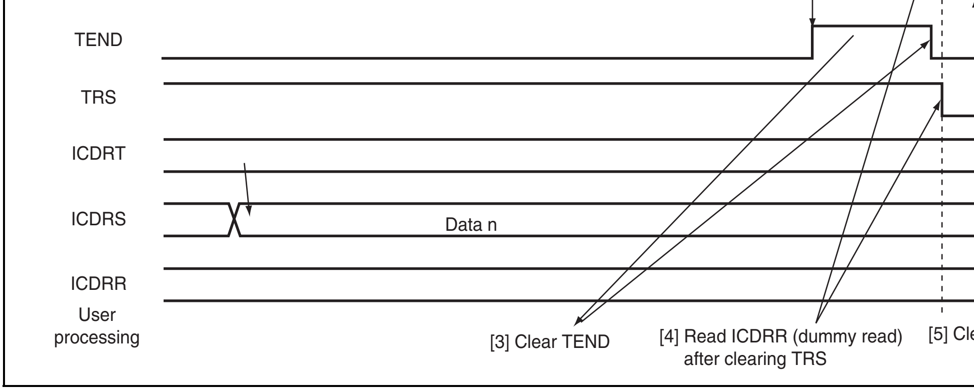


**Figure 16.10 Slave Transmit Mode Operation Timing (2)**

the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise
clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (
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
fixed low until ICDRR is read. The change of the acknowledge before reading ICDR
returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

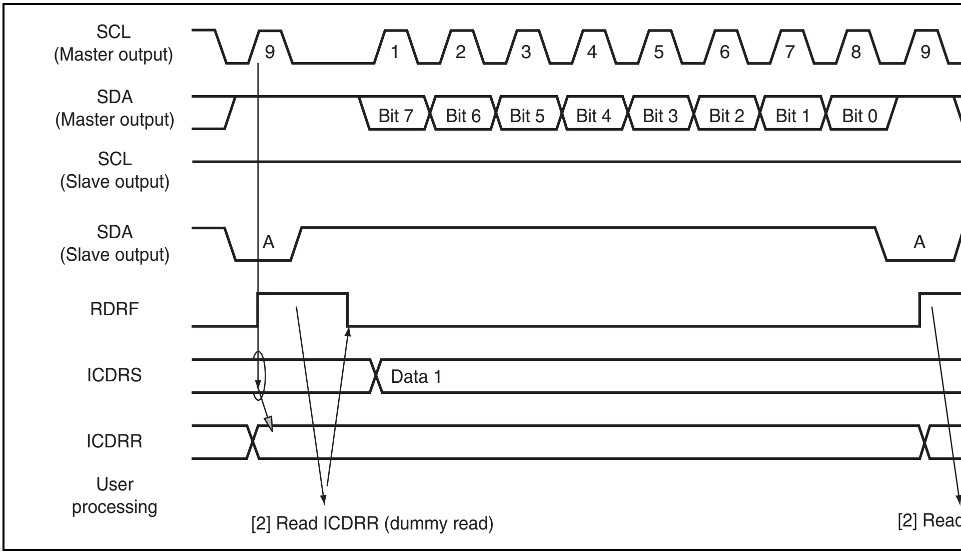


**Figure 16.11 Slave Receive Mode Operation Timing (1)**

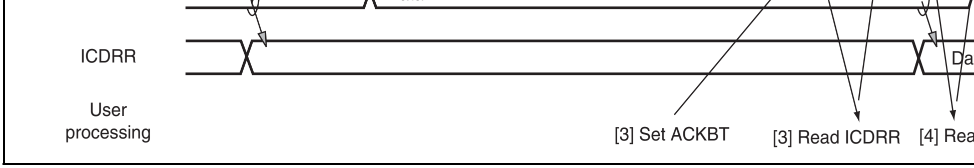


**Figure 16.12 Slave Receive Mode Operation Timing (2)**

### 16.4.6 Clock Synchronous Serial Format

This module can be operated with the clock synchronous serial format, by setting the FS
SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selecte
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 ris
of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in e
MSB first or LSB first. The output level of SDA can be changed during the transfer wait,
SDAO bit in ICCR2.

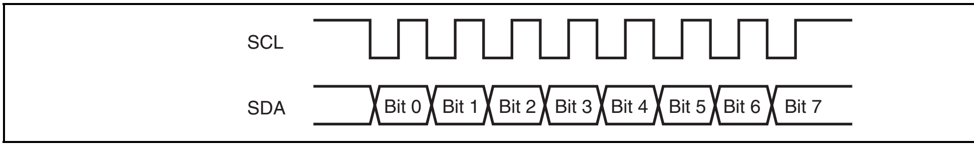


**Figure 16.13 Clock Synchronous Serial Transfer Format**

transmission is performed by writing data to ICDRT every time TDRE is set. When from transmit mode to receive mode, clear TRS while TDRE is 1.

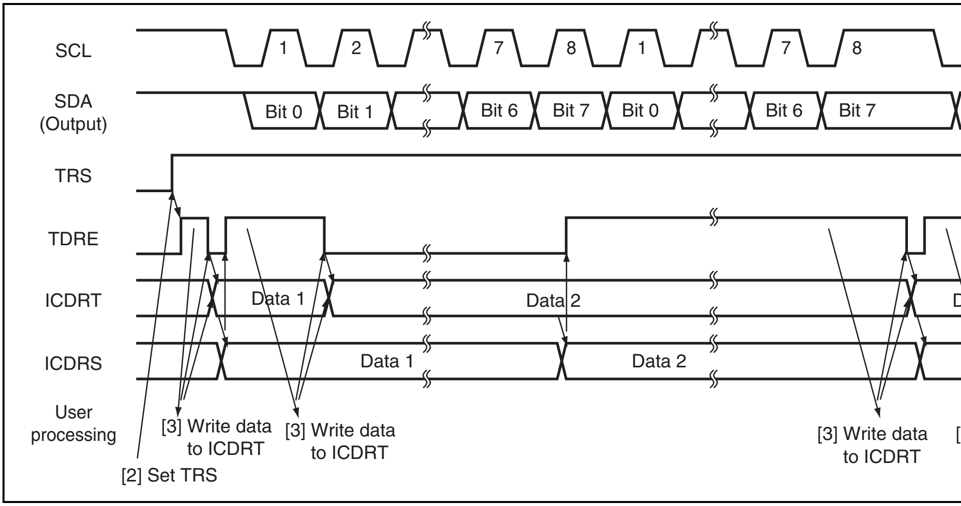


**Figure 16.14 Transmit Mode Operation Timing**

RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected anc AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRI

4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, fixed high after receiving the next byte data.

Notes: Follow the steps below to receive only one byte with MST=1 specified. See figur for the operation timing.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS3 to CKS0 in ICCR1. (Initial settir
2. Set MST=1 while the RCVD bit in ICCR1 is 0. This causes the receive clock output.
3. Check if the BC2 bit in ICMR is set to 1 and then set the RCVD bit in ICCR1 This causes the SCL to be fixed to the high level after outputting one byte of t receive clock.

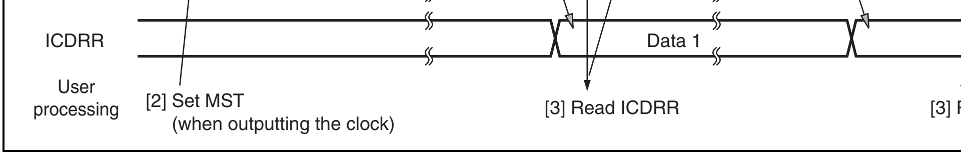


**Figure 16.15 Receive Mode Operation Timing**

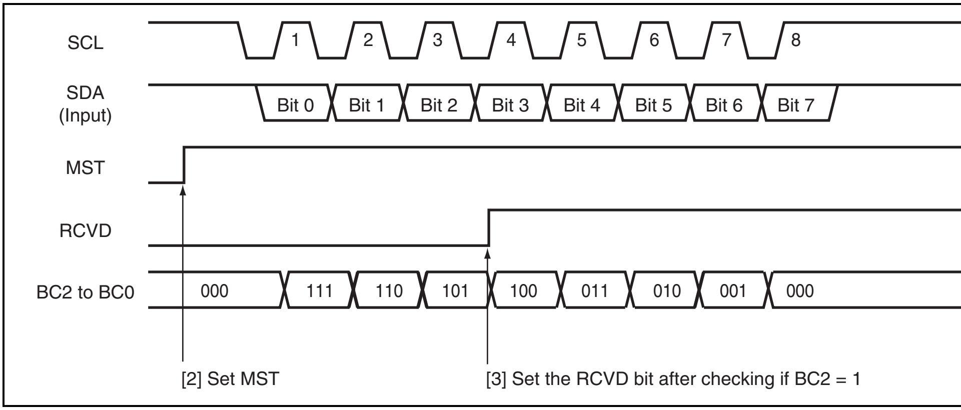


**Figure 16.16 Operation Timing For Receiving One Byte**

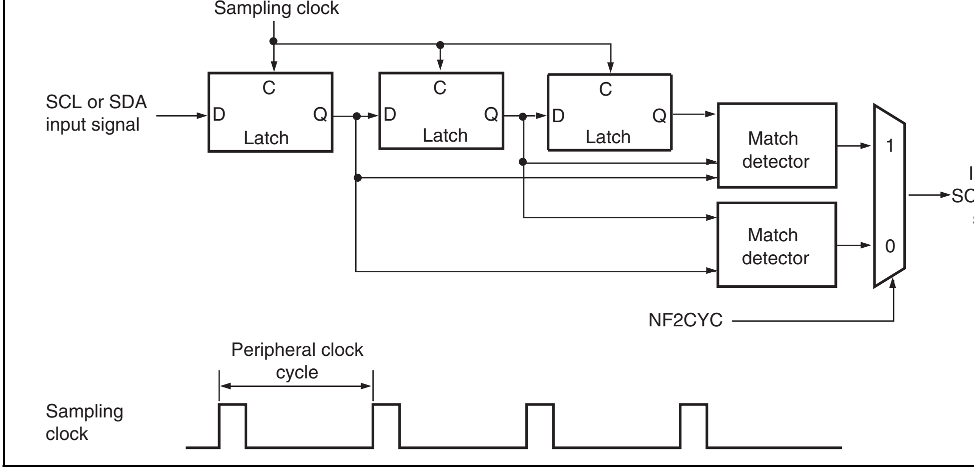
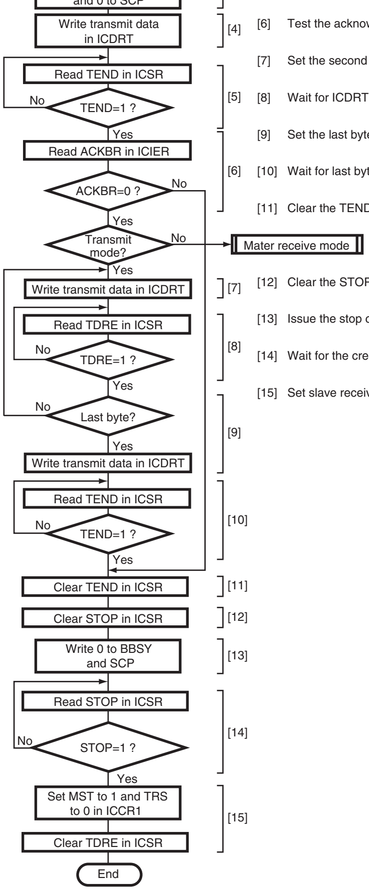


Figure 16.17 Block Diagram of Noise Filter



[6] Test the acknowledge transferred from the specified slave device.

[7] Set the second and subsequent bytes (except for the final byte) of transmit data.

[8] Wait for ICDRT empty.

[9] Set the last byte of transmit data.

[10] Wait for last byte to be transmitted.

[11] Clear the TEND flag.

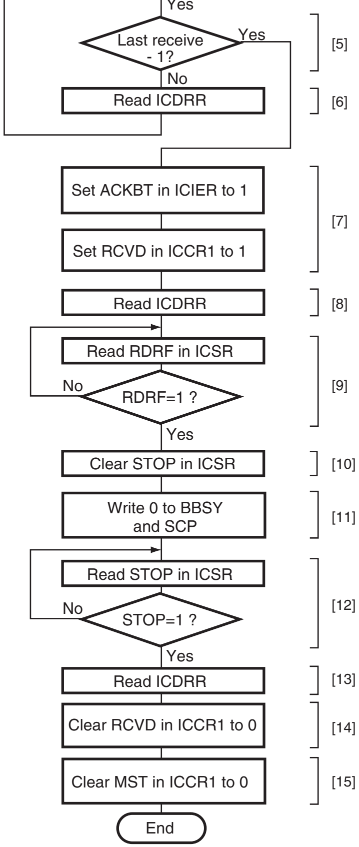
[12] Clear the STOP flag.

[13] Issue the stop condition.

[14] Wait for the creation of stop condition.

[15] Set slave receive mode. Clear TDRE.

**Figure 16.18 Sample Flowchart for Master Transmit Mode**



[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.

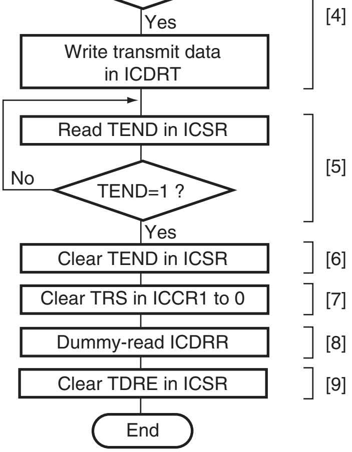
[14] Clear RCVD.

[15] Set slave receive mode.

Notes: * Make sure that no interrupt will be generated during steps [1] to

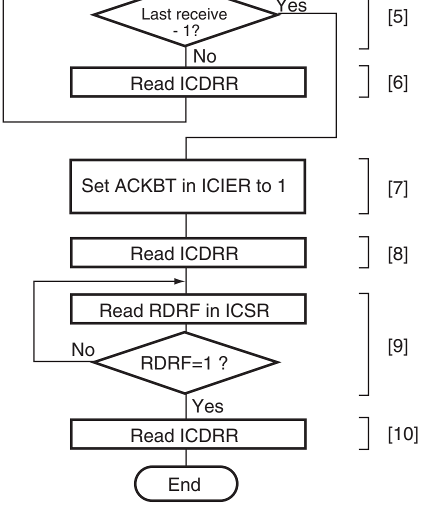
When the size of receive data is only one byte in reception, steps [2] to [6] are skipped after step [1], before jumping to step
The step [8] is dummy-read in ICDRR.
However, when the size of receive data is two bytes and more, steps [2] to [6] are not skipped after step [1].

**Figure 16.19 Sample Flowchart for Master Receive Mode**



[9] Clear the TDRE flag.

**Figure 16.20 Sample Flowchart for Slave Transmit Mode**



[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 byt reception, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [ is dummy-read in ICDRR.
However, when the size of receive data is two bytes and more, steps [2] to [6] are not skippe after step [1].

**Figure 16.21 Sample Flowchart for Slave Receive Mode**

| | | | | | |
|---|---|---|---|---|---|
| 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 handling. Interrupt sources should be cleared in the exception handling. The TDRE and bits are automatically cleared to 0 by writing the transmit data to ICDRT. The RDRF bit automatically cleared to 0 by reading ICDRR. The TDRE bit is set to 1 again at the sam when the transmit data is written to ICDRT. Therefore, when the TDRE bit is cleared to 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; the TDRE, TEND, and RDRF bits are not cleared automatically when the transfer is cor

| Item | Master Transmit Mode | Master Receive Mode | Slave Transmit Mode | Slave Re Mode |
|---|---|---|---|---|
| Slave address + $R/\overline{W}$ bit transmission/ reception | Transmission by DTC (ICDR write) | Transmission by CPU (ICDR write) | Reception by CPU (ICDR read) | Receptio (ICDR rea |
| 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) | Receptio (ICDR rea |
| Last frame processing | Not necessary | Reception by CPU (ICDR read) | Not necessary | Receptio (ICDR rea |
| 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 | Receptio data cour |

Figure 16.22 shows the timing of the bit synchronous circuit and table 16.6 shows the ti
SCL output changes from low to Hi-Z then SCL is monitored.

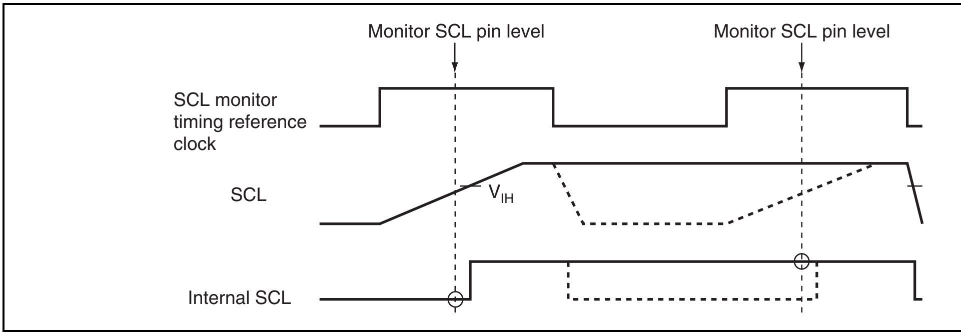


**Figure 16.22 The Timing of the Bit Synchronous Circuit**

| | |
|---|---|
| 1 | 39.5 $t_{pcyc}$*2 |

Notes: 1. SCL pin level is monitored after "time for monitoring SCL" has elapsed from the edge of the reference clock for monitoring SCL.

2. $t_{pcyc}$ indicates the period of the peripheral clock.

is recognized. The fall of the ninth clock pulse can be recognized by checking the SCLO
I$^2$C bus control register 2 (ICCR2). When a stop condition or repeated start condition is
specific timing under the conditions 1 or 2 shown below, the condition may not be outpu
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 Synch 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 ei 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 sto condition are to be issued in sequence, be sure to transmit a slave address before issuing condition.

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 IC cannot be read before the rising edge of the 8th clock so that the next round of reception p with the RDRF bit in ICSR set to 1, the 8the clock is fixed low and the 9th clock is outpu

If ICDRR cannot be read before the rising edge of the 8th clock of SCL, set the RCVD bi ICRR1 to 1 so that transfer proceeds in byte units.

### 16.8.6 Supported Emulator

The E200F emulator does not support $I^2C2$ operation. Use the E10A emulator when debu $I^2C2$ operation.

- Fast A/D conversion

  When operating at Pϕ = 40 MHz, conversion time is 1.25 µs per channel (A/D clock 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_ makes a total of twelve 16-bit A/D data registers (ADDR). The SH7132 and SH713 eight registers for both A/D_0 and A/D_1, which makes a total of sixteen 16-bit A/D registers (ADDR). A/D conversion results are stored in A/D data registers (ADDR) t 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 sample simultaneously because sample-and-hold circuits can be dedicated for channels 0 to 10.

  — Group A (GrA): Analog input pins selected from channels 0, 1, and 2 can be simultaneously sampled.

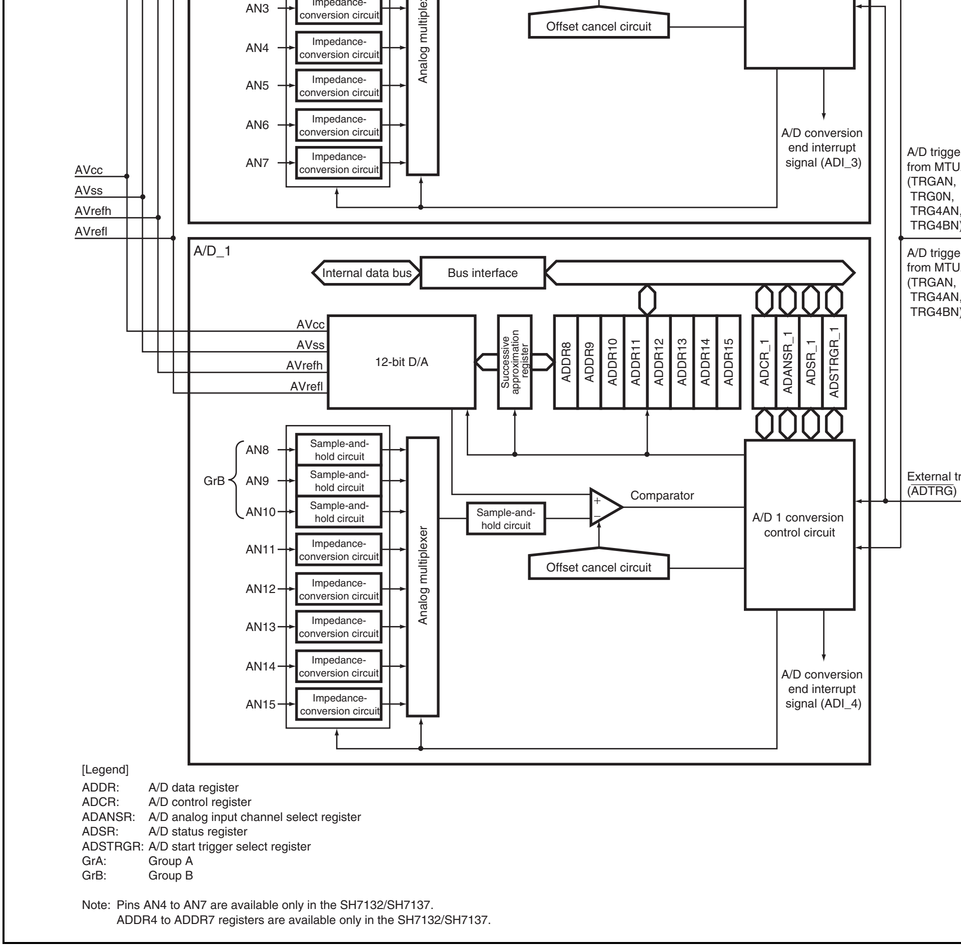
  — Group B (GrB): Analog input pins selected from channels 8, 9, and 10 can be simultaneously sampled.

- 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)


AN3
Impedance-conversion circuit
AN4
Impedance-conversion circuit
AN5
Impedance-conversion circuit
AN6
Impedance-conversion circuit
AN7
Impedance-conversion circuit
Analog multiple
Offset cancel circuit
A/D conversion end interrupt signal (ADI_3)
A/D trigge from MTU (TRGAN, TRG0N, TRG4AN, TRG4BN)
AVcc
AVss
AVrefh
AVrefl
A/D_1
Internal data bus
Bus interface
A/D trigge from MTU (TRGAN, TRG4AN, TRG4BN)
AVcc
AVss
AVrefh
AVrefl
12-bit D/A
Successive approximation register
ADDR8
ADDR9
ADDR10
ADDR11
ADDR12
ADDR13
ADDR14
ADDR15
ADCR_1
ADANSR_1
ADSR_1
ADSTRGR_1
GrB
AN8
AN9
AN10
Sample-and-hold circuit
Sample-and-hold circuit
Sample-and-hold circuit
AN11
AN12
AN13
AN14
AN15
Impedance-conversion circuit
Impedance-conversion circuit
Impedance-conversion circuit
Impedance-conversion circuit
Impedance-conversion circuit
Analog multiplexer
Sample-and-hold circuit
Comparator
+
−
Offset cancel circuit
A/D 1 conversion control circuit
External tr (ADTRG)
A/D conversion end interrupt signal (ADI_4)

[Legend]

| | |
|---|---|
| ADDR: | A/D data register |
| ADCR: | A/D control register |
| ADANSR: | A/D analog input channel select register |
| ADSR: | A/D status register |
| ADSTRGR: | A/D start trigger select register |
| GrA: | Group A |
| GrB: | Group B |

Note: Pins AN4 to AN7 are available only in the SH7132/SH7137.
ADDR4 to ADDR7 registers are available only in the SH7132/SH7137.

**Figure 17.1 Block Diagram of A/D Converter**

|  |  |  |  |  |
|---|---|---|---|---|
|  | $AV_{SS}$ | 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) ($AV_{refl} < AV_{refh}$) | √ |
|  | $\overline{\text{ADTRG}}$ | Input | A/D external trigger input pin | √ |
| A/D module 0 (A/D_0) | AN0 | Input | Analog input pin 0 (Group A) | √ |
|  | AN1 | Input | Analog input pin 1 (Group A) | √ |
|  | AN2 | Input | Analog input pin 2 (Group A) | √ |
|  | AN3 | Input | Analog input pin 3 | √ |
|  | 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 | √ |

| | | | | | |
|---|---|---|---|---|---|
| register_0 | | | | | |
| 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 |

| | | | | |
|---|---|---|---|---|
| 7 | ADST | 0 | R/W | A/D Start<br>When this bit is cleared to 0, A/D conversion is st and the A/D converter enters the idle state. Wher is set to 1, A/D conversion is started. In single-cy mode, this bit is automatically cleared to 0 when conversion ends on the selected single channel. continuous scan mode, A/D conversion is continu performed for the selected channels in sequence bit is cleared by software, a reset, software stand or module standby mode. |
| 6 | ADCS | 0 | R/W | A/D Continuous Scan<br>Selects either a single-cycle or a continuous scar mode. This bit is valid only when scan mode is se<br>0: Single-cycle scan<br>1: Continuous scan<br>When changing the operating mode, first clear th bit to 0. |
| 5 | ACE | 0 | R/W | Automatic Clear Enable<br>Enables or disables the automatic clearing of AD ADDR is read by the CPU or DTC. When this bit to 1, ADDR is automatically cleared to H'0000 aft CPU or DTC reads ADDR. This function allows th 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 |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | 0: Generation of A/D conversion end interrupt is<br>1: Generation of A/D conversion end interrupt is |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value always be 0. |
| 1 | TRGE | 0 | R/W | Trigger Enable<br>Enables or disables A/D conversion start by the trigger input ($\overline{\text{ADTRG}}$) or A/D conversion start tr from the MTU2 and MTU2S (TRGAN, TRG0N, and TRG4BN from the MTU2 and TRGAN, TRG TRG4BN from the MTU2S). For selection of the trigger and A/D conversion start trigger from the MTU2S, see the description of the EXTRG bit.<br>0: A/D conversion start by the external trigger or conversion start trigger from the MTU or MTU disabled<br>1: A/D conversion start by the external trigger or conversion start trigger from the MTU2 or MT enabled |

conversion start by the external trigger input is er
only when the ADST bit is cleared to 0.

When the external trigger is used as an A/D conv
start trigger, the low-level pulse input to the $\overline{\text{ADTR}}$
must be at least 1.5 Pϕ clock cycles in width.

0: A/D converter is started by the A/D conversion
trigger from the MTU2 or MTU2S

1: A/D converter is started by the external pin ($\overline{\text{AD}}$

| 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 valu<br>always be 0. |
| 0 | ADF | 0 | R/(W)* | A/D End Flag<br>A status flag that indicates the completion of A/<br>conversion.<br>[Setting condition]<br>• When A/D conversion on all specified chan<br>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 interr<br>ADDR is read |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value sho always be 0. |
| 6 | STR6 | 0 | R/W | Start Trigger 6<br>Enables or disables the A/D conversion start rec input from the MTU2S.<br>0: Disables the A/D conversion start by TRGAN (MTU2S).<br>1: Enables the A/D conversion start by TRGAN (MTU2S). |
| 5 | STR5 | 0 | R/W | Start Trigger 5<br>Enables or disables the A/D conversion start rec input from the MTU2S.<br>0: Disables the A/D conversion start by TRG4AN (MTU2S).<br>1: Enables the A/D conversion start by TRG4AN (MTU2S). |
| 4 | STR4 | 0 | R/W | Start Trigger 4<br>Enables or disables the A/D conversion start rec input from the MTU2S.<br>0: Disables the A/D conversion start by TRG4BN (MTU2S).<br>1: Enables the A/D conversion start by TRG4BN (MTU2S). |

| | | | | |
|---|---|---|---|---|
| | | | | Enables or disables the A/D conversion start re<br>input from the MTU2.<br>0: Disables the A/D conversion start by TRGAN<br>(MTU2).<br>1: Enables the A/D conversion start by TRGAN<br>(MTU2). |
| 1 | STR1 | 0 | R/W | Start Trigger 1<br>Enables or disables the A/D conversion start re<br>input from the MTU2.<br>0: Disables the A/D conversion start by TRG4A<br>(MTU2).<br>1: Enables the A/D conversion start by TRG4A<br>(MTU2). |
| 0 | STR0 | 0 | R/W | Start Trigger 0<br>Enables or disables the A/D conversion start re<br>input from the MTU2.<br>0: Disables the A/D conversion start by TRG4E<br>(MTU2).<br>1: Enables the A/D conversion start by TRG4B<br>(MTU2). |

| Bit | Bit Name | Value | R/W | Description |
|---|---|---|---|---|
| 7 | ANS7 | 0 | R/W | Setting bits in the A/D analog input channel sele register to 1 selects a channel that corresponds specified bit. For the correspondence between a input pins and bits, see table 17.3.<br>When changing the analog input channel, the A 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 |

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

| 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**

| A/D_0 Converter | | A/D_1 Converter | |
|---|---|---|---|
| Analog Input Channels | A/D Data Registers | Analog Input Channels | A/D Data Reg |
| 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 |

**Figure 17.2 Interface Between CPU and 12-Bit A/D Converter**

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 completed, the ADF bit in ADSR is set to 1 and the ADST bit is automatically cleared t continuous scan mode, when conversion on all specified channels is completed, the AD ADSR is set to 1. To stop A/D conversion, write 0 to the ADST bit. When the ADF bit i if the ADIE bit in ADCR is set to 1, an A/D conversion end interrupt (ADI) is generated clearing the ADF bit to 0, read the ADF bit while set to 1 and then write 0. However, w 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 selected and the A/D_0 conversion is performed in single-cycle scan mode using four c 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 (ADAN 1.
3. Set the ADST bit in the A/D control register_0 (ADCR_0) to 1 to start A/D conversi
4. After channels 0 to 2 (GrA) are sampled simultaneously, offset canceling processing performed. Then, A/D conversion is performed on channel 0. Upon completion of th conversion, the A/D conversion result is transferred to ADDR0. Following this, cha converted. Upon completion of the conversion, the A/D conversion result is transfer ADDR1. In the same way, channel 2 is converted and the A/D conversion result is tr to ADDR2.

   A/D conversion of channel 3 is then started. Upon completion of the A/D conversio conversion result is transferred to ADDR3.

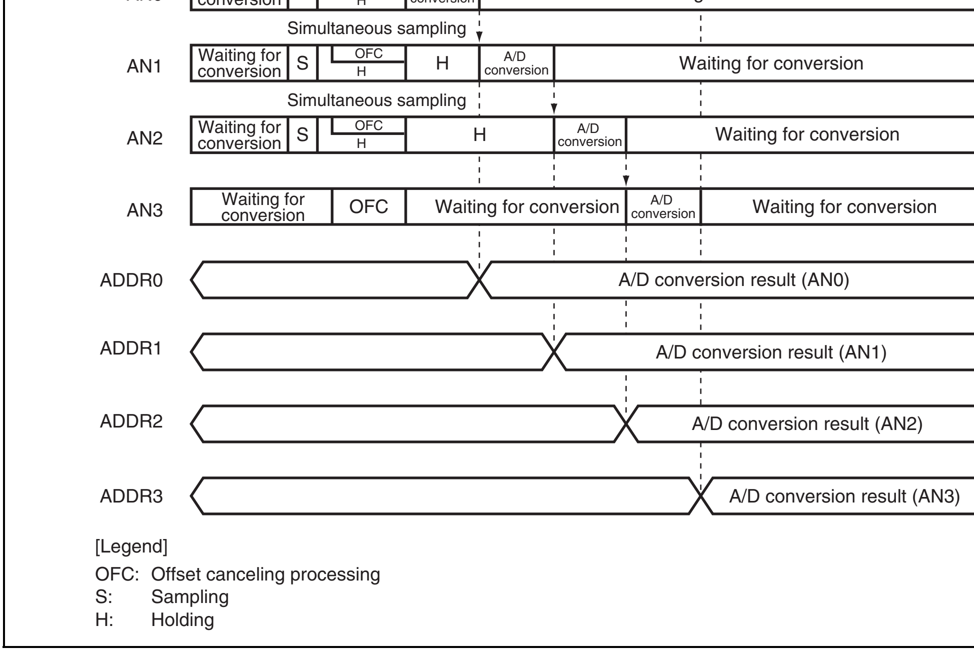
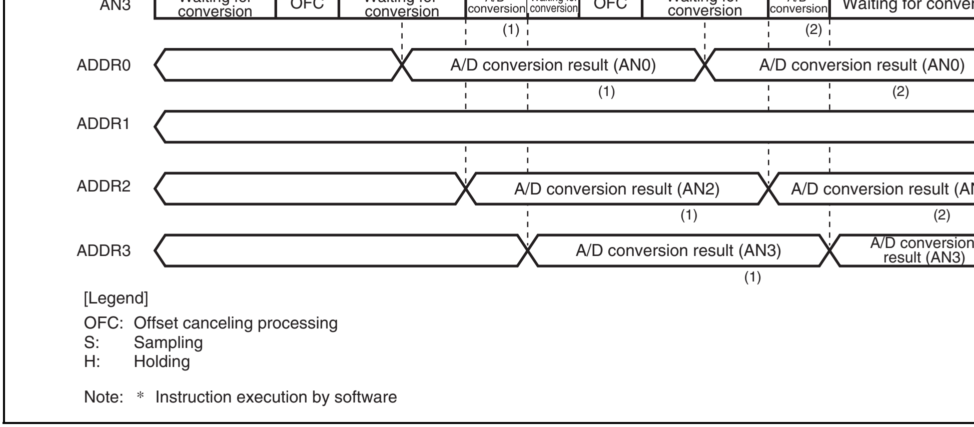


**Figure 17.3 Example of A/D_0 Converter Operation (Single-Cycle Scan Mo**

0, channel 1 is not sampled. After this, offset canceling processing (OFC) is perform
the A/D conversion on channel 0 is started. Upon completion of the A/D conversion,
conversion result is transferred to ADDR0. In the same way, channel 2 is converted
A/D conversion result is transferred to ADDR2. The A/D conversion is not performe
channel 1.

5. The A/D conversion of channel 3 is started. Upon completion of the A/D conversion
conversion result is transferred to ADDR3.
6. When the A/D conversion ends on all the specified channels (AN0 to AN3), the ADI
to 1. At this time, if the ADIE bit is set to 1, an ADI_3 interrupt is generated after th
conversion.
7. Steps 4 to 6 are repeated as long as the ADST bit remains set to 1. When the ADST
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.



[Legend]

OFC: Offset canceling processing
S: Sampling
H: Holding

Note: * Instruction execution by software

**Figure 17.4 Example of A/D_0 Converter Operation (Continuous Scan Mod**

When an event that sets the ADST bit writing to this bit by the CPU, A/D converter acti request from the MTU2, the MTU2S, and an external trigger signal occurs, the analog in sampled by the dedicated sample-and-hold circuit for each channel after the A/D conver delay time ($t_D$) has passed and the offset canceling processing (OFC) is performed. After sampling of the analog input using the sample-and-hold circuit common to all the chann performed and then the A/D conversion is started. Figure 17.5 shows the A/D conversio this case. This A/D conversion time ($t_{CONV}$) includes the $t_D$, the offset canceling processin ($t_{OFC}$), the analog input sampling time with a dedicated sample-and-hold circuit for each ($t_{SPLSH}$), and the analog input sampling time with the sample-and-hold circuit common to channels ($t_{SPL}$). The $t_{SPLSH}$ does not depend on the number of channels simultaneously sam

In continuous scan mode, the A/D conversion time ($t_{CONV}$) given in table 17.6 applies to t conversion time of the first cycle. The conversion time of the second and subsequent cy expressed as ($t_{CONV} - t_D + 6$).

| | | | |
|---|---|---|---|
| AN4 | — | AN12 | — |
| AN5 | — | AN13 | — |
| AN6 | — | AN14 | — |
| AN7 | — | AN15 | — |

**Table 17.6 A/D Conversion Time**

| | | Number of Required Stat | | |
|---|---|---|---|---|
| **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 |

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)

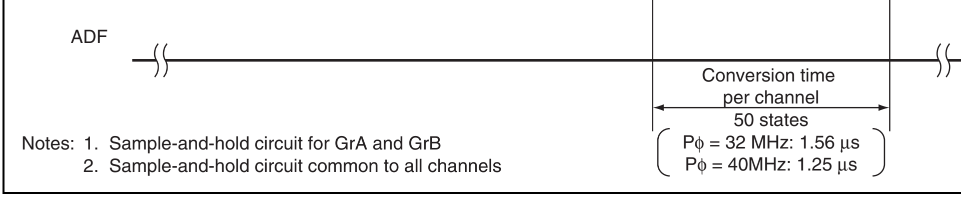

ADF
Conversion time
per channel
50 states
Pϕ = 32 MHz: 1.56 µs
Pϕ = 40MHz: 1.25 µs
Notes: 1. Sample-and-hold circuit for GrA and GrB
2. Sample-and-hold circuit common to all channels

**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, TR
TRG4BN) from the MTU2 and A/D conversion start triggers (TRGAN, TRG4AN, and
from the MTU2S. To enable this function, set the TRGE bit in ADCR to 1 and clear the
bit to 0. After this setting is made, if an A/D conversion start trigger from the MTU2 or
generated, the ADST bit is set to 1. The timing between the setting of the ADST bit and
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
have been set.

the ADCR, ADSTRGR, and ADANSR registers have been set.

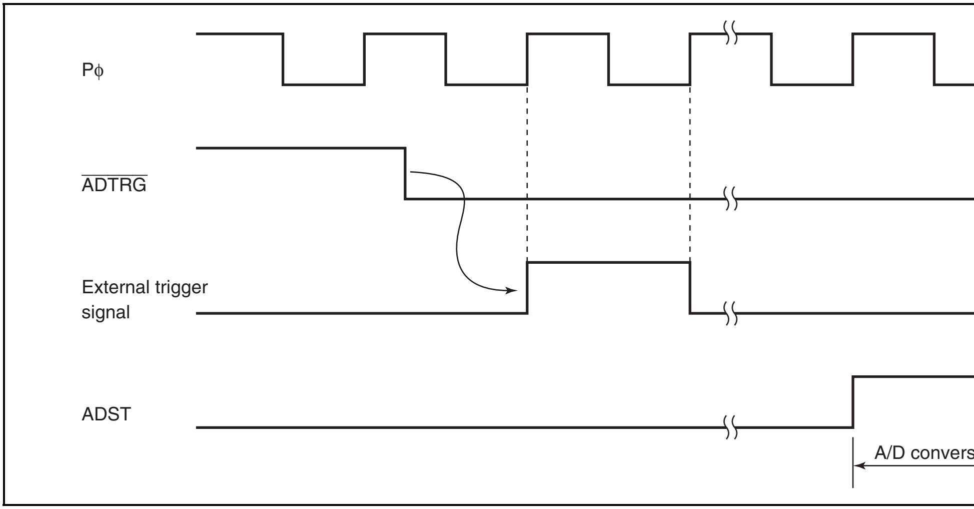


**Figure 17.6 External Trigger Input Timing**

### 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 automati cleared to H'0000 by setting the ACE bit in ADCR to 1. This function allows the detectio ADDR renewal failure.

Figure 17.7 shows an example of when the auto-clear function of ADDR is disabled (nor and enabled.

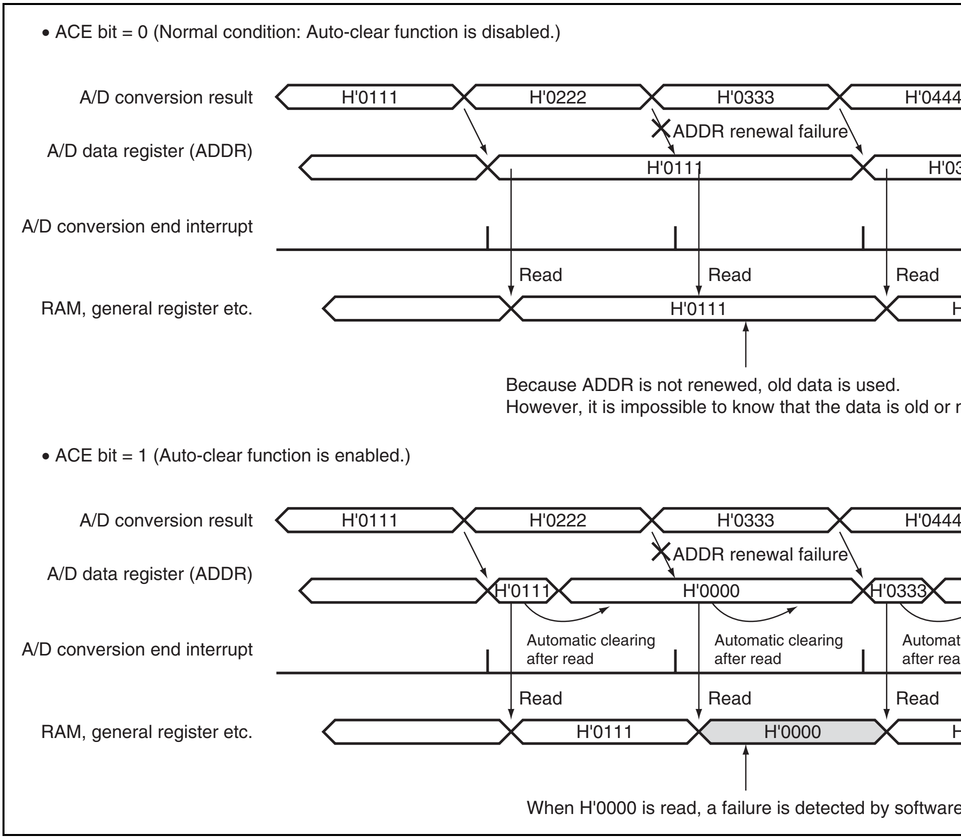

• ACE bit = 0 (Normal condition: Auto-clear function is disabled.)
A/D conversion result
H'0111
H'0222
H'0333
H'0444
ADDR renewal failure
A/D data register (ADDR)
H'0111
H'03
A/D conversion end interrupt
Read
Read
Read
RAM, general register etc.
H'0111
H
Because ADDR is not renewed, old data is used.
However, it is impossible to know that the data is old or n
• ACE bit = 1 (Auto-clear function is enabled.)
A/D conversion result
H'0111
H'0222
H'0333
H'0444
ADDR renewal failure
A/D data register (ADDR)
H'0111
H'0000
H'0333
Automatic clearing after read
Automatic clearing after read
Automat after rea
A/D conversion end interrupt
Read
Read
Read
RAM, general register etc.
H'0111
H'0000
H
When H'0000 is read, a failure is detected by software

**Figure 17.7 Example of When ADDR Auto-clear Function is Disabled (Normal Condition)/Enabled**

**Table 17.7 Interrupt Sources**

| Channel | Interrupt Source | Interrupt Enable Bit | Interrupt Flag Bit | DTC Activation | P |
|---|---|---|---|---|---|
| A/D_0 | A/D_3 | ADIE | ADF | Possible | H |
| A/D_1 | A/D_4 | ADIE | ADF | Possible | L |

figure 17.8).

- Full-scale error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conver characteristic when the digital output value changes from B'111111111110 to the ma voltage value (full-scale voltage) B'111111111111. Does not include a quantization 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 conver characteristic between zero voltage and full-scale voltage. Does not include offset er scale error, or quantization error (see figure 17.8).

- Absolute accuracy

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

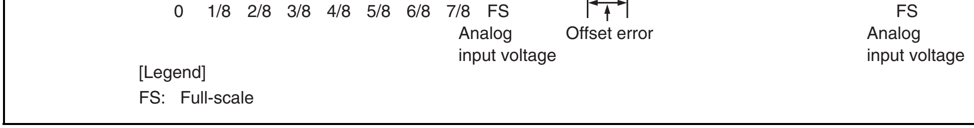


**Figure 17.8 Definitions of A/D Conversion Accuracy**

### 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 $AV_{refh} \leq$ AVcc. If these conditions are not met, the reliability of the LSI may be adversel 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 p and the layout in which the digital circuit signal lines and analog circuit signal lines cros close proximity to each other should be avoided as much as possible. Failure to do so m the incorrect operation of the analog circuitry due to inductance, adversely affecting the conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN15), an reference power supply ($AV_{refh}$ and $AV_{refl}$), the analog power supply (AVcc), and the ana (AVss). Also, AVss should be connected at one point to a stable digital ground (Vss) on

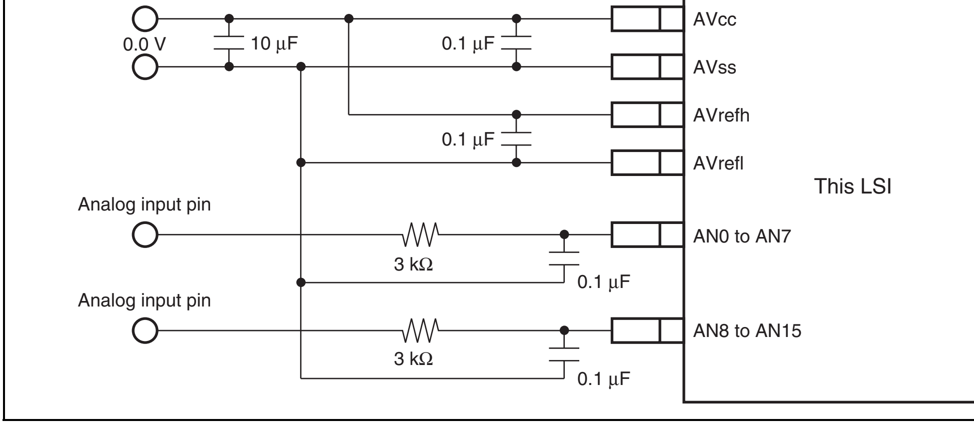


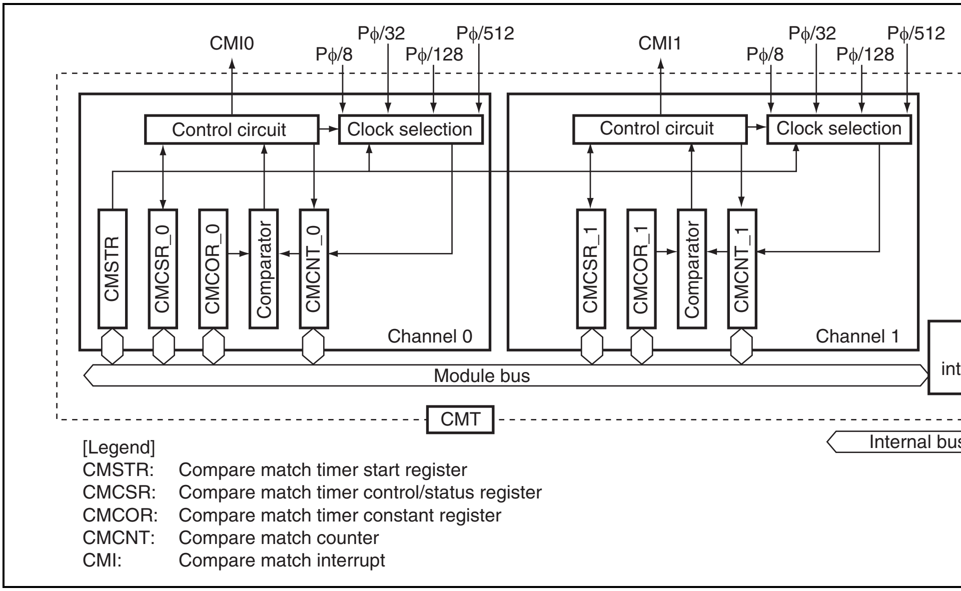
**Figure 17.9 Example of Analog Input Pin Protection Circuit**

### 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 (ADSTRGR) and the A/D analog input channel select register (ADANSR) have been not modify the settings of the ADCS, ACE, ADIE, TRGE, and EXTRG bits while the 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 inp channel select register (ADANSR) are all 0.

- Interrupt request on compare match
- Module standby mode can be set.

Figure 18.1 shows a block diagram of CMT.



[Legend]

| | |
|---|---|
| CMSTR: | Compare match timer start register |
| CMCSR: | Compare match timer control/status register |
| CMCOR: | Compare match timer constant register |
| CMCNT: | Compare match counter |
| CMI: | Compare match interrupt |

**Figure 18.1 Block Diagram of CMT**

| | | | | | |
|---|---|---|---|---|---|
| Compare match timer control/status register_1 | CMCSR_0 | R/W | H'0000 | H'FFFFCE02 | 8, 1 |
| Compare match counter_0 | CMCNT_0 | R/W | H'0000 | H'FFFFCE04 | 8, 1 |
| Compare match constant register_0 | CMCOR_0 | R/W | H'FFFF | H'FFFFCE06 | 8, 1 |
| Compare match timer control/status register_0 | CMCSR_1 | R/W | H'0000 | H'FFFFCE08 | 8, 1 |
| Compare match counter_1 | CMCNT_1 | R/W | H'0000 | H'FFFFCE0A | 8, 1 |
| Compare match constant register_1 | CMCOR_1 | R/W | H'FFFF | H'FFFFCE0C | 8, 1 |

| | | | | |
|---|---|---|---|---|
| 15 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va<br>always be 0. |
| 1 | STR1 | 0 | R/W | Count Start 1<br>Specifies whether compare match counter 1 o<br>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 o<br>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 the counter input clock.

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

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

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | When 0 is written to this bit after reading C<br>• When CMT registers are accessed when th of the DISEL bit of MRB in the DTC is 0 aft 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 ( generation when CMCNT and CMCOR values (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 valu 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 fou clocks obtained by dividing the peripheral oper clock (Pϕ). When the STR bit in CMSTR is set CMCNT starts counting on the clock selected v 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 w

2. he flag is set by another compare match before writing 0 to the bit after readin the flag will not be cleared by writing 0 to it once. In this case, read the bit as 1 and write 0 to it.

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

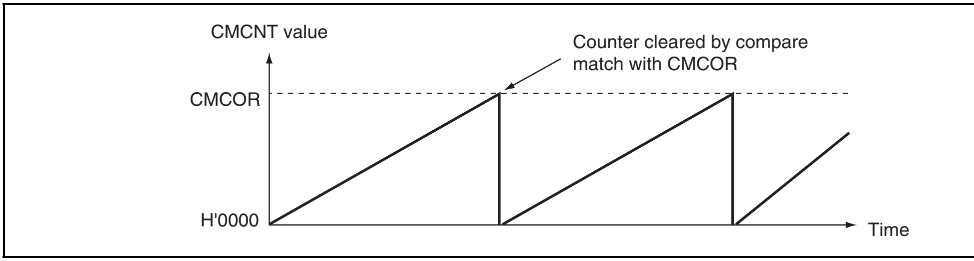


**Figure 18.2 Counter Operation**

### 18.3.2 CMCNT Count Timing

One of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the P
can be selected with bits CKS1 and CKS0 in CMCSR. Figure 18.3 shows the timing.

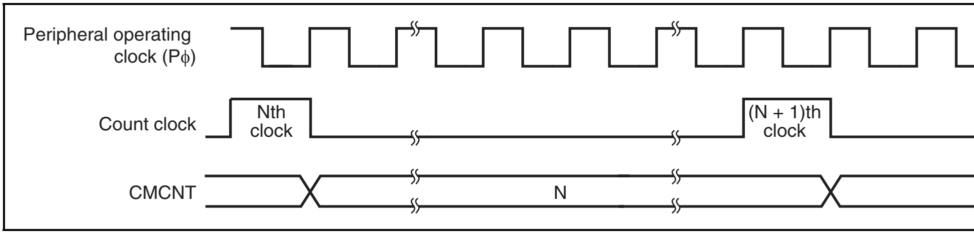


**Figure 18.3 Count Timing**

priority between channels is fixed. See section 8, Data Transfer Controller (DTC), for de

Table 18.2 lists the CMT interrupt sources.

**Table 18.2 Interrupt Source**

| Channel | Interrupt Source | Interrupt Enable Bit | Interrupt Flag Bit | DTC Activation |
|---|---|---|---|---|
| 0 | CMI_0 | CMIE | CMF | Possible |
| 1 | CMI_1 | CMIE | CMF | Possible |

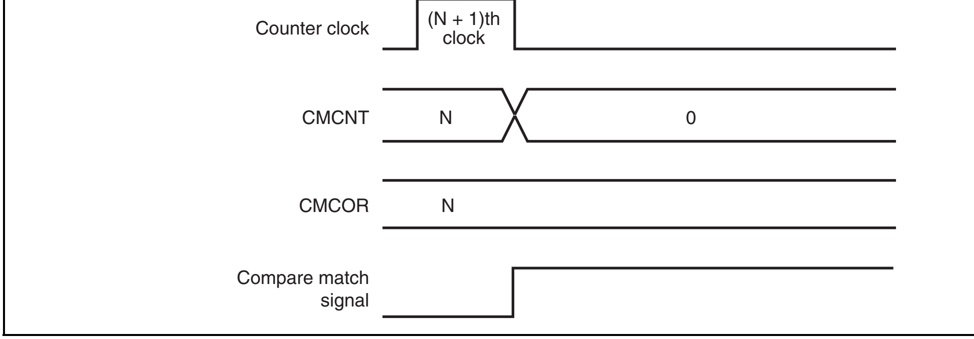


**Figure 18.4 Timing of CMF Setting**

### 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.

CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 18
the timing to clear the CMCNT counter.

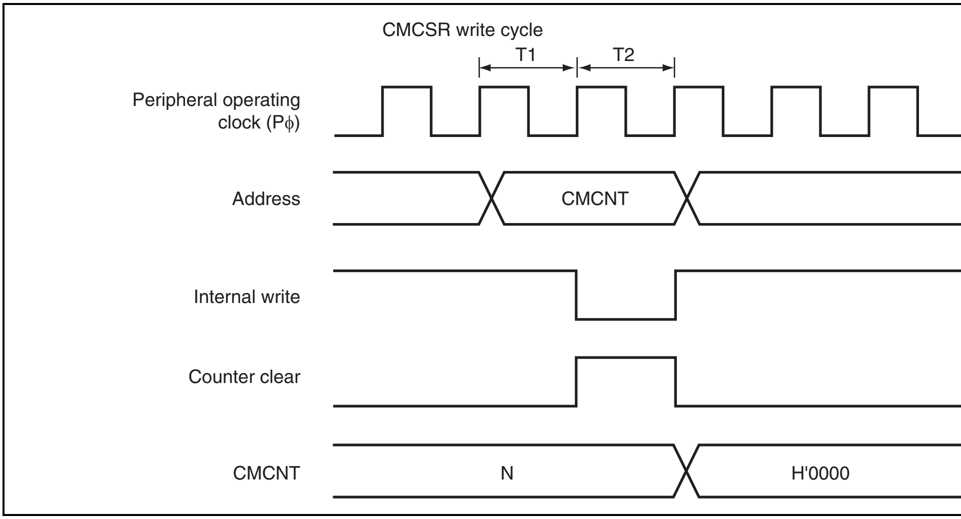


**Figure 18.5 Conflict between Write and Compare-Match Processes of CMC**

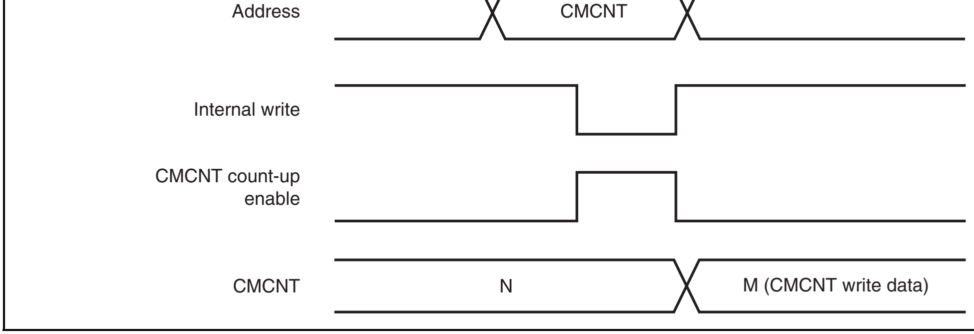


Figure 18.6 Conflict between Word-Write and Count-Up Processes of CMCN

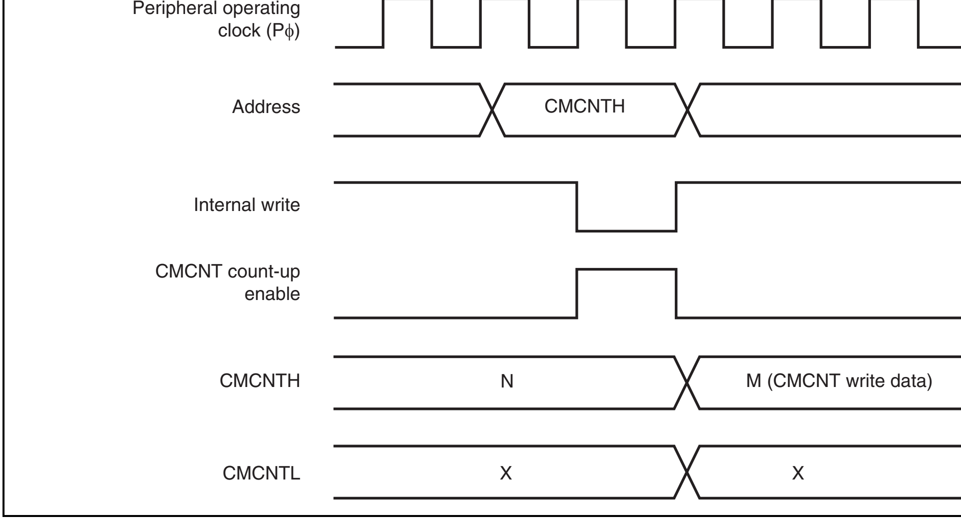


**Figure 18.7 Conflict between Byte-Write and Count-Up Processes of CMC**

### 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 se
CMF bit in CMCSR is set to 1 and CMCNT is cleared to H'0000.

### 19.1.2 Scope

The CAN Data Link Controller function is not described in this document. It is the respo of the reader to investigate the CAN Specification Document (see references). The inter the CAN Controller are described, in so far as they pertain to the connection with the Us Interface.

The programming model is described in some detail. It is not the intention of this docum describe the implementation of the programming interface, but to simply present the int the underlying CAN functionality.

The document places no constraints upon the implementation of the RCAN-ET module process, packaging or power supply criteria. These issues are resolved where appropriat implementation specifications.

### 19.1.3 Audience

In particular this document provides the design reference for software authors who are r 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.

### 19.1.5 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 c 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 proble priority inversion for real-time applications
- data buffer access without SW handshake requirement in reception
- flexible micro-controller interface
- flexible interrupt structure

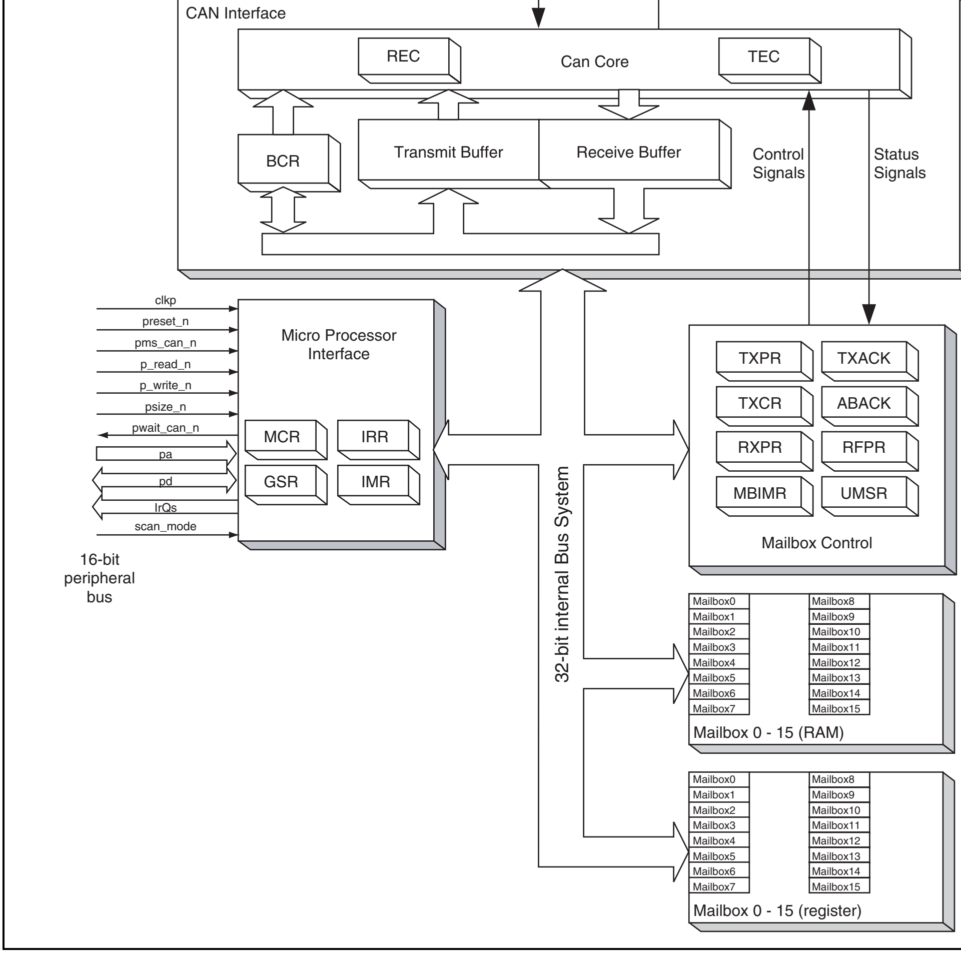


**Figure 19.1 RCAN-ET Architecture**

- Mailbox

  The Mailboxes consists of RAM configured as message buffers and registers. There a 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 store messages from the CAN Interface into the Mailbox and set/clear appropriate accordingly.

  — To transmit messages, RCAN-ET will run the internal arbitration to pick the corre priority message, and load the message from the Mailbox into the Tx-buffer of the 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, UMS MBIMR.

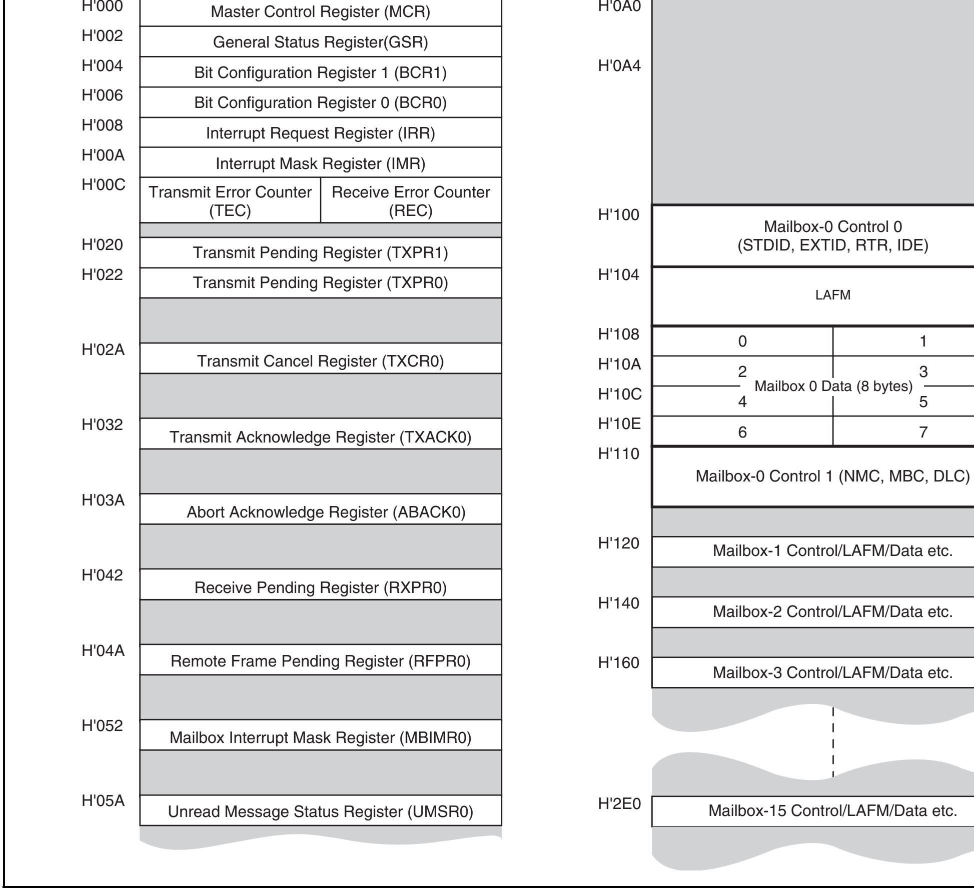

H'000 Master Control Register (MCR)
H'002 General Status Register(GSR)
H'004 Bit Configuration Register 1 (BCR1)
H'006 Bit Configuration Register 0 (BCR0)
H'008 Interrupt Request Register (IRR)
H'00A Interrupt Mask Register (IMR)
H'00C Transmit Error Counter (TEC) Receive Error Counter (REC)
H'020 Transmit Pending Register (TXPR1)
H'022 Transmit Pending Register (TXPR0)
H'02A Transmit Cancel Register (TXCR0)
H'032 Transmit Acknowledge Register (TXACK0)
H'03A Abort Acknowledge Register (ABACK0)
H'042 Receive Pending Register (RXPR0)
H'04A Remote Frame Pending Register (RFPR0)
H'052 Mailbox Interrupt Mask Register (MBIMR0)
H'05A Unread Message Status Register (UMSR0)
H'0A0
H'0A4
H'100 Mailbox-0 Control 0 (STDID, EXTID, RTR, IDE)
H'104 LAFM
H'108 0 1
H'10A 2 3
H'10C 4 5
H'10E 6 7
Mailbox 0 Data (8 bytes)
H'110 Mailbox-0 Control 1 (NMC, MBC, DLC)
H'120 Mailbox-1 Control/LAFM/Data etc.
H'140 Mailbox-2 Control/LAFM/Data etc.
H'160 Mailbox-3 Control/LAFM/Data etc.
H'2E0 Mailbox-15 Control/LAFM/Data etc.

**Figure 19.2 RCAN-ET Memory Map**

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

| | | | | |
|---|---|---|---|---|
| 0 (Receive Only) | 100 – 103 | 104 – 107 | 108 – 10F | 110 – |
| 1 | 120 – 123 | 124 – 127 | 128 – 12F | 130 – |
| 2 | 140 – 143 | 144 – 147 | 148 – 14F | 150 – |
| 3 | 160 – 163 | 164 - 167 | 168 – 16F | 170 – |
| 4 | 180 – 183 | 184 – 187 | 188 – 18F | 190 – |
| 5 | 1A0 – 1A3 | 1A4 – 1A7 | 1A8 – 1AF | 1B0 – |
| 6 | 1C0 – 1C3 | 1C4 – 1C7 | 1C8 – 1CF | 1D0 – |
| 7 | 1E0 – 1E3 | 1E4 – 1E7 | 1E8 – 1EF | 1F0 – |
| 8 | 200 – 203 | 204 – 207 | 208 – 20F | 210 – |
| 9 | 220 – 223 | 224 – 227 | 228 – 22F | 230 – |
| 10 | 240 – 243 | 244 – 247 | 248 – 24F | 250 – |
| 11 | 260 – 263 | 264 – 267 | 268 – 26F | 270 – |
| 12 | 280 – 283 | 284 – 287 | 288 – 28F | 290 – |
| 13 | 2A0 – 2A3 | 2A4 – 2A7 | 2A8 – 2AF | 2B0 – |
| 14 | 2C0 – 2C3 | 2C4 – 2C7 | 2C8 – 2CF | 2D0 – |
| 15 | 2E0 – 2E3 | 2E4 – 2E7 | 2E8 – 2EF | 2F0 – |

Mailbox-0 is a receive-only box, and all the other Mailboxes can operate as both receive
transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message
The following diagram shows the structure of a Mailbox in detail.

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H'104 + N*32 | LAFM | 0 | 0 | | | | | | | | | | | | 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 | | | | | | | | | | | | | | | | Access Size | Field Nam |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| H'100 + N*32 | IDE | RTR | 0 | STDID[10:0] | | | | | | | | | | | EXTID[17:16] | | Word/LW | Control 0 |
| H'102 + N*32 | EXTID[15:0] | | | | | | | | | | | | | | | | Word | |
| H'104 + N*32 | IDE_LAFM | 0 | 0 | STDID_LAFM[10:0] | | | | | | | | | | | EXTID_LAFM[17:16] | | Word/LW | LAFM |
| H'106 + N*32 | EXTID_LAFM[15:0] | | | | | | | | | | | | | | | | Word | |
| H'108 + N*32 | MSG_DATA_0 (first Rx/Tx Byte) | | | | | | | | MSG_DATA_1 | | | | | | | | Byte/Word/LW | Data |
| H'10A + N*32 | MSG_DATA_2 | | | | | | | | MSG_DATA_3 | | | | | | | | Byte/Word | |
| H'10C + N*32 | MSG_DATA_4 | | | | | | | | MSG_DATA_5 | | | | | | | | Byte/Word/LW | |
| H'10E + N*32 | MSG_DATA_6 | | | | | | | | MSG_DATA_7 | | | | | | | | Byte/Word | |
| H'110 + N*32 | 0 | 0 | NMC | ATX | DART | MBC[2:0] | | | 0 | 0 | 0 | 0 | DLC[3:0] | | | | Byte/Word | Control 1 |

**Figure 19.3 Mailbox-N Structure**

Notes: 1. All bits shadowed in grey are reserved and must be written LOW. The value r 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 Ma limited.

3. ID Reorder (MCR15) can change the order of STDID, RTR, IDE and EXTID message control and LAFM.

corresponding RFPR set or IRR[2] (Remote Frame Request Interrupt), however, as RCA 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) and ATX=1 the RTR flag must be programmed to zero to allow data frame to be transm

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 ext format of CAN data frames and remote frames.

| IDE | Description |
|---|---|
| 0 | Standard format |
| 1 | Extended format |

R/W: R R R/W R/W R/W R/W R/W R/W R R R R R/W R/W

**NMC (New Message Control):** When this bit is set to '0', the Mailbox of which the RXP RFPR bit is already set does not store the new message but maintains the old one and sets UMSR correspondent bit. When this bit is set to '1', the Mailbox of which the RXPR or R 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 ve be that both RXPR and RFPR flags (together with UMSR) are set for the same Mailbox. 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 Remot is received into the Mailbox DLC is stored. Then, a Data Frame is transmitted from the sa 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 I priority or Mailbox priority as configured with the Message Transmission Priority contro (MCR.2). In order to use this function, MBC[2:0] needs to be programmed to be '001' (Bi a transmission is performed by this function, the DLC (Data Length Code) to be used is t 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 u ID of remote frame has to be exactly the same as that of data frame as the reply message.

| 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 auto
transmission of a message in the event of an error on the CAN bus or an arbitration lost
CAN bus. In effect, when this function is used, the corresponding TXCR bit is automati
the start of transmission. When this bit is set to '0', RCAN-ET tries to transmit the messa
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
When MBC=111 (Bin), the Mailbox is inactive, i.e., it does not receive or transmit a me
regardless of TXPR or other settings. The MBC='110', '101' and '100' settings are prohib
When the MBC is set to any other value, the LAFM field becomes available. Please don
TXPR when MBC is set as reception. There is no hardware protection, and TXPR remai
MBC[1] of Mailbox-0 is fixed to "1" by hardware. This is to ensure that MB0 cannot be
configured to transmit Messages.

- LAFM can be use

| | | | |
|---|---|---|---|
| 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=00
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, will be transmitted in a data frame. Please note that when a remote frame request is transm 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 |

**Figure 19.4 Acceptance Filter**

If a bit is set in the LAFM, then the corresponding bit of a received CAN identifier is ig when the RCAN-ET searches a Mailbox with the matching CAN identifier. If the bit is c then the corresponding bit of a received CAN identifier must match to the STDID/IDE/ in the mailbox to be stored. The structure of the LAFM is same as the message control i 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 Ma As soon as RCAN-ET finds one matching, it stops the search. The message will be store depending on the NMC and RXPR/RFPR flags. This means that, even using LAFM, a re message can only be stored into 1 Mailbox.

**Important:** When a message is received and a matching Mailbox is found, the whole m stored into the Mailbox. This means that, if the LAFM is used, the STDID, RTR, IDE ar may differ to the ones originally set as they are updated with the STDID, RTR, IDE and 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" |

| | |
|---|---|
| 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] corres the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 thr bit 0.

| | | | |
|---|---|---|---|
| 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-

- MCR (Address = H'000)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MCR15 | MCR14 | - | - | - | TST[2:0] | | | MCR7 | MCR6 | MCR5 | - | - | MCR2 |
| Initial value: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 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 |

**Bit 15 — ID Reorder (MCR15):** This bit changes the order of STDID, RTR, IDE and both message control and LAFM.

| Bit15 : MCR15 | Description |
|---|---|
| 0 | RCAN-ET is the same as HCAN2 |
| 1 | RCAN-ET is not the same as HCAN2 (Initial value) |

**Figure 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 × 1<br>Recessive Bits) (Initial value) |
| 1 | RCAN-ET moves directly into Halt Mode after it enters BusOff  if MC<br>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 n before activating the Test Mode it is requested to move RCAN-ET into Halt mode or Res This is to avoid that the transition to Test Mode could affect a transmission/reception in p 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 RC used in normal operation.

setting prohibited

**Bit 7 — Auto-wake Mode (MCR7):** MCR7 enables or disables the Auto-wake mode. I set, the RCAN-ET automatically cancels the sleep mode (MCR5) by detecting CAN bus (dominant bit). If MCR7 is cleared the RCAN-ET does not automatically cancel the slee

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 o mode. Please note that when Halt is entered in Bus Off the CAN engine is also recoverin immediately to Error Active mode.

| Bit6 : MCR6 | Description |
|---|---|
| 0 | If MCR[1] is set, RCAN-ET will not enter Halt mode during Bus Off up to end of recovery sequence (Initial value) |
| 1 | Enter Halt mode immediately during Bus Off if MCR[1] or MCR[14] asserted. |

method is used, RCAN-ET will miss the first message to receive. CAN transceivers stanc
mode will also be unable to cope with the first message when exiting stand by mode, and
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
That allows the CPU to clear all pending interrupts before entering sleep mode. Once all
are cleared RCAN-ET must leave the Halt mode and enter Sleep mode simultaneously (b
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 transmiss
pending transmit data. If this bit is set, pending transmit data are sent in order of the bit p
the Transmission Pending Register (TXPR). The order of transmission starts from Mailbo
the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for
transmission).

| | |
|---|---|
| 1 | Transmission order determined by mailbox number priority (Mailbo<br>Mailbox-1) |

**Bit 1 — Halt Request (MCR1):** Setting the MCR1 bit causes the CAN controller to co current operation and then enter Halt mode (where it is cut off from the CAN bus). The remains in Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interf not join the CAN bus activity and does not store messages or transmit messages. All the registers (including Mailbox contents and TEC/REC) remain unchanged with the except IRR0 and GSR4 which are used to notify the halt status itself. If the CAN bus is in idle intermission state regardless of MCR6, RCAN-ET will enter Halt Mode within one Bit MCR6 is set, a halt request during Bus Off will be also processed within one Bit Time. the full Bus Off recovery sequence will be performed beforehand. Entering the Halt Mo notified by IRR0 and GSR4.

If both MCR14 and MCR6 are set, MCR1 is automatically set as soon as RCAN-ET ent BusOff.

In the Halt mode, the RCAN-ET configuration can be modified with the exception of th Timing setting, as it does not join the bus activity. MCR[1] has to be cleared by writing order to re-join the CAN bus. After this bit has been cleared, RCAN-ET waits until it de 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 cle until the transition to Halt mode is completed (notified by IRR0 and GSR4). Aft is set this can be cleared only after entering Halt mode or through a reset operati HW).

Note: Transition into or recovery from HALT mode, is only possible if the BCR1 and registers are configured to a proper Baud Rate.

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 v
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 h
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) |

**Bits 15 to 6: Reserved.** The written value should always be '0' and the returned value is

**Bit 5 — Error Passive Status Bit (GSR5):** Indicates whether the CAN Interface is in E Passive or not. This bit will be set high as soon as the RCAN-ET enters the Error Passiv is cleared when the module enters again the Error Active state (this means the GSR5 wil during Error Passive and during Bus Off). Consequently to find out the correct state bot 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 Passi Mode is selected |

**Bit 4 — Halt/Sleep Status Bit (GSR4):** Indicates whether the CAN engine is in the hal state or not. Please note that the clearing time of this flag is not the same as the setting t IRR12.

Please note that this flag reflects the status of the CAN engine and not of the full RCAN RCAN-ET exits sleep mode and can be accessed once MCR5 is cleared. The CAN engi 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 inter idle or MCR5 is set and RCAN-ET is in the halt mode or RCAN-ET to Bus Off when MCR14 and MCR6 are both set |

is set at the 7th bit of End Of Frame. GSR2 is set at the 3rd bit of intermission if there are n messages ready to be transmitted. It is also set by arbitration lost, bus idle, reception, rese 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 (In 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 |
| 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 a requested by the Bus Off recovery sequence. However the flag GSR1 is not set in

**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 re (Initial value) |
| 1 | [Setting condition] When TEC ≥ 256 (bus off state) |

Note: Only the lower 8 bits of TEC are accessible from the user interface. The 9th bit is ec to GSR0.

used peripheral bus frequency.

- BCR1 (Address = H'004)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TSG1[3:0] | | | | - | TSG2[2:0] | | | - | - | SJW[1:0] | | - | - |
| Initial value: | 0 | 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 |

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 segment TSEG1 (= PRSEG + PHSEG1) to compensate for edges on the CAN Bus with phase error. A value from 4 to 16 time quanta can be set.

| Bit 15:<br>TSG1[3] | Bit 14:<br>TSG1[2] | Bit 13:<br>TSG1[1] | Bit 12:<br>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'.

| | | | |
|---|---|---|---|
| 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

**Bits 5 and 4 — ReSynchronisation Jump Width (SJW[1:0] = BCR0[5:4]):** These bits 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 |

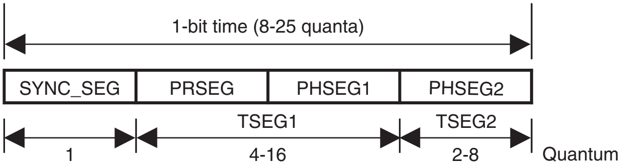
**Bits 3 to 1: Reserved**. The written value should always be '0' and the returned value is '0'

**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 PHSEG1) |

| BRP[7] | BRP[6] | BRP[5] | BRP[4] | BRP[3] | BRP[2] | BRP[1] | BRP[0] | Description |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 X peripheral bu<br>(Initial value) |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 X peripheral bu |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 6 X peripheral bu |
| :<br>: | :<br>: | :<br>: | :<br>: | :<br>: | :<br>: | :<br>: | :<br>: | 2*(register value<br>peripheral bus c |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 512 X peripheral |

- Requirements of Bit Configuration Register



SYNC_SEG: Segment for establishing synchronisation of nodes on the CAN bus. (N
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 ext
when synchronisation (resynchronisation) is established.)

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

TSEG1: TSG1 + 1

$f_{CLK}$ = 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 allow

TSEG2 ≥ 2

These constraints allow the setting range shown in the table below for TSEG1 and TSEG Bit Configuration Register. The number in the table shows possible setting of SJW. "No" that there is no allowed combination of TSEG1 and TSEG2.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **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 poss
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 s
BPR = 4, TSEG1 = 8, TSEG2 = 5.

Then the configuration to write is BCR1 = 7400 and BCR0 = 0004.

**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 n 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 a present. While the RCAN-ET is in sleep mode and a dominant bit is detected on the CAN bit is set. This interrupt is cleared by writing a '1' to this bit position. Writing a '0' has no auto wakeup is not used and this interrupt is not requested it needs to be disabled by the r interrupt mask register. If auto wake up is not used and this interrupt is requested it shoul cleared only after recovering from sleep mode. This is to avoid that a new falling edge of 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 GS

| 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 i mode |

| Bit 9: IRR9 | Description |
|---|---|
| 0 | No pending notification of message overrun/overwrite<br>[Clearing condition] Clearing of all bit in UMSR/setting MBIMR for a set (initial value) |
| 1 | A receive message has been discarded due to overrun condition or message has been overwritten<br>[Setting condition] Message is received while the corresponding RX and/or RFPR =1 and MBIMR =0 |

**Bit 8 — Mailbox Empty Interrupt Flag (IRR8):** This bit is set when one of the messa 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 clear this mailbox is now ready to accept a new message data for the next transmission. In eff bit is set by an OR'ed signal of the TXACK and ABACK bits not masked by the corresp MBIMR flag. Therefore, this bit is automatically cleared when all the TXACK and AB are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBI Writing to this bit position has no effect.

| Bit 8: IRR8 | Description |
|---|---|
| 0 | Messages set for transmission or transmission cancellation reques progressed. (Initial value)<br>[Clearing Condition] All the TXACK and ABACK bits are cleared/se MBIMR for all TXACK and ABACK set |
| 1 | Message has been transmitted or aborted, and new message can<br>[Setting condition]<br>When one of the TXPR bits is cleared by completion of transmissio completion of transmission abort, i.e., when a TXACK or ABACK bi MBIMR=0). |

**Bit 6 — Bus Off Interrupt Flag (IRR6):** This bit is set when RCAN-ET enters the Bus- or when RCAN-ET leaves Bus-off and returns to Error-Active. The cause therefore is the 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). T remains set even if the RCAN-ET node leaves the bus-off condition, and needs to be expl cleared by S/W. The S/W is expected to read the GSR0 to judge whether RCAN-ET is in off or error active status. It is cleared by writing a '1' to this bit position even if the node i 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 retu from Bus-off<br>[Setting condition] When TEC becomes ≥ 256 or End of Bus-off afte consecutive recessive bits or transition from Bus Off to Halt |

**Bit 5 — Error Passive Interrupt Flag (IRR5):** Interrupt flag indicating the error passiv caused by the transmit or receive error counter or by Error Passive forced by test mode. T reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the may still be error passive. Please note that the SW needs to check GSR0 and GSR5 to jud 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 t mode is used |

transmit error counter (TEC) reaches a value greater than 95. The interrupt is reset by w
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 been received in a mailbox. This bit is set if at least one receive mailbox, with related M set, contains a remote frame transmission request. This bit is automatically cleared whe the Remote Frame Receive Pending Register (RFPR), are cleared. It is also cleared by w 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 correspo MBIMR = 0 |

**Bit 1 — Data Frame Received Interrupt Flag (IRR1):** IRR1 indicates that there are p Data Frames received. If this bit is set at least one receive mailbox contains a pending m This bit is cleared when all bits in the Data Frame Receive Pending Register (RXPR) ar i.e. there is no pending message in any receiving mailbox. It is in effect a logical OR of flags from each configured receive mailbox with related MBIMR not set. It is also clear writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no e

3. Sleep mode has been entered after a sleep request (MCR5) has been made while in Ha

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 Sleep mode. So, IRR0 is not set if RCAN-ET enters Halt mode again right after exiting fr mode, without GSR4 being cleared. Similarly, IRR0 is not set by direct transition from S mode to Halt Request. At the transition from Halt/Sleep mode to Transition/Reception, cl 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 s 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 mode (Initial value)<br>[Setting condition] When reset/halt/sleep transition is completed afte (MCR0 or HW) or Halt mode (MCR1) or Sleep mode (MCR5) is requ |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 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 |

**Bits 15 to 0:** Maskable interrupt sources corresponding to IRR[15:0] respectively. Whe set, the interrupt signal is not generated, although setting the corresponding IRR bit is st performed.

| Bit[15:0]: IMRn | Description |
|---|---|
| 0 | Corresponding IRR is not masked (IRQ is generated for interrupt c |
| 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/(w register that functions as a counter indicating the number of transmit/receive message er CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [1 and [4]. When not in (Write Error Counter) test mode this register is read only, and can modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) entering to bus off.

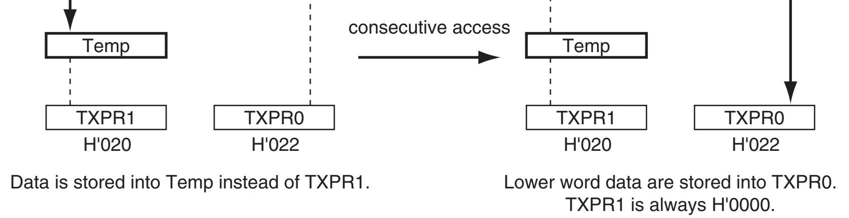
In Write Error Counter test mode (i.e. TST[2:0] = 3'b100), it is possible to write to this The same value can only be written to TEC/REC, and the value written into TEC is set and REC. When writing to this register, RCAN-ET needs to be put into Halt Mode. Thi only intended for test purposes.

The following sections describe RCAN-ET Mailbox registers that control / flag individua Mailboxes. The address is mapped as follows.

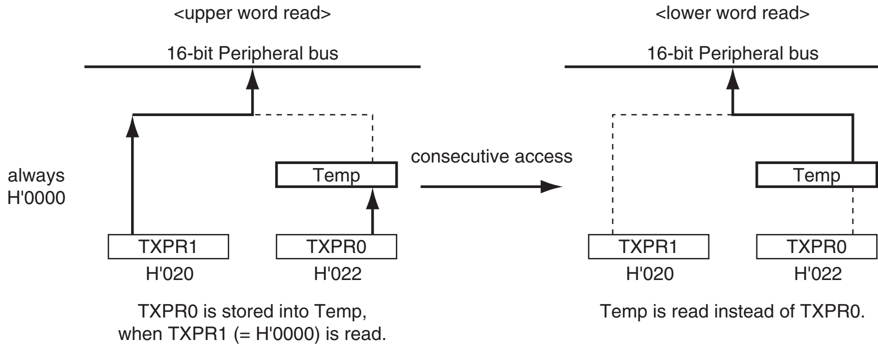
**Important:** LongWord access is carried out as two consecutive Word accesses.

|  |  |  |  |
|---|---|---|---|
| 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**



<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 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 M except for the Mailbox-0. Writing a '1' to a bit location when the mailbox is not configure transmit is not allowed.

even under circumstances such as bus arbitration losses or errors on the CAN bus. Pleas
section 19.4, Application Note.

When the RCAN-ET changes the state of any TXPR bit position to a '0', an empty slot i
(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 i
TXACK register, and if a message transmission abortion is successful it is signalled in t
ABACK register. By checking these registers, the contents of the Message of the corres
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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TXPR1[15:0] | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | 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 T
no effect.

- TXPR0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TXPR0[15:1] | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |

Note: * it is possible only to write a '1' for a Mailbox configured as transmitter.

**Bit 0 — Reserved**. This bit is always '0' as this is a receive-only Mailbox. Writing a '1' to
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
Mailbox-1.This register is used by the CPU to request the pending transmission requests
TXPR to be cancelled. To clear the corresponding bit in the TXPR the CPU must write a
bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXC
and sets the corresponding ABACK bit. However, once a Mailbox has started a transmiss
cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the
controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXAC
however, if the transmission fails due to a bus arbitration loss or an error on the bus, the C
controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABAC
an attempt is made by the CPU to clear a mailbox transmission that is not transmit-pendi
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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TXCR0[15:1] | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | 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 config
transmit.

**Bit 0**: This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit posi 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 sig CPU that a mailbox transmission has been successfully made. When a transmission has the RCAN-ET sets the corresponding bit in the TXACK register. The CPU may clear a ' 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TXACK0[15:1] | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | 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 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 (Dat Remote Frame)<br>[Setting Condition] Completion of message transmission for corresp mailbox |

**Bit 0**: This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit posi no effect and always read back as a '0'.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ABACK0[15:1] | | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W: | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R |

Note: * Only when writing a '1' to clear.

**Bits 15 to 1**: Notifies that the requested transmission cancellation of the corresponding M
has been performed successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respecti

| Bit[15:1]:ABACK0 | Description |
|---|---|
| 0 | [Clearing Condition] Writing '1' (Initial value) |
| 1 | Corresponding Mailbox has cancelled transmission of message (Da<br>Remote Frame)<br>[Setting Condition] Completion of transmission cancellation for corre<br>mailbox |

**Bit 0**: This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit positi
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 co
the received Data Frames pending flags associated with the configured Receive Mailboxe
a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is se
RXPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing
no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mai
Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data F
Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and
interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by
Data Frames and not by receiving Remote frames.

| | |
|---|---|
| 0 | [Clearing Condition] Writing '1' (Initial value) |
| 1 | Corresponding Mailbox received a CAN Data Frame<br>[Setting Condition] Completion of Data Frame receive on correspor mailbox |

### (6) Remote Frame Receive Pending Register (RFPR0)

The RFPR0 is a 16-bit read / conditionally-write registers. The RFPR is a register that c received Remote Frame pending flags associated with the configured Receive Mailboxe CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing no effect. In effect there is a bit position for all mailboxes. However, the bit may only be mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. RFPR bit is set, it also sets IRR2 (Remote Frame Request Interrupt Flag) if its MBIMR Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not s note that these bits are only set by receiving Remote Frames and not by receiving Data f

- RFPR0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFPR0[15:0] | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | 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.

The MBIMR1 and MBIMR0 are 16-bit read/write registers. The MBIMR only prevents
setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received In
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 th
corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR
IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or
Similarly when a mailbox has been configured for transmission, a mask prevents the gene
an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmiss
abortion of transmission (IRR[8]), however, it does not prevent the RCAN-ET from clear
corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and
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
masked. At reset all mailbox interrupts are masked.

- MBIMR0

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MBIMR0[15:0] | | | | | | | | | | | | | |
| Initial value: | 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 |

**Bits 15 to 0:** Enable or disable interrupt requests from individual Mailbox-15 to Mailbox
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) |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | UMSR0[15:0] | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W* | R/W* | 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 c 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 ove condition<br>[Setting Condition] When a new message is received before RXPR is cleared |

| 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 the values, and the Tx Output is disabled so that RCAN-ET does generate error frames or acknowledgment bits. IRR13 is set whe message error occurs.

Self Test Mode 1: RCAN-ET generates its own Acknowledge bit, and can store its messages into a reception mailbox (if required). The Rx/Tx pins connected to the CAN bus.

Self Test Mode 2: RCAN-ET generates its own Acknowledge bit, and can store its messages into a reception mailbox (if required). The Rx/Tx pins need to be connected to the CAN bus or any external devices, as internal Tx is looped back to the internal Rx. Tx pin outputs only recessive bits and Rx pin is disabled.

Note: the REC will not be modified by implementing this Mod
However, once running in Error Passive Mode, the REC will in
normally should errors be received. In this Mode, RCAN-ET w
BusOff if TEC reaches 256 (Dec). However when this mode is
RCAN-ET will not be able to become Error Active. Consequen
end of the Bus Off recovery sequence, RCAN-ET will move to
Passive and not to Error Active

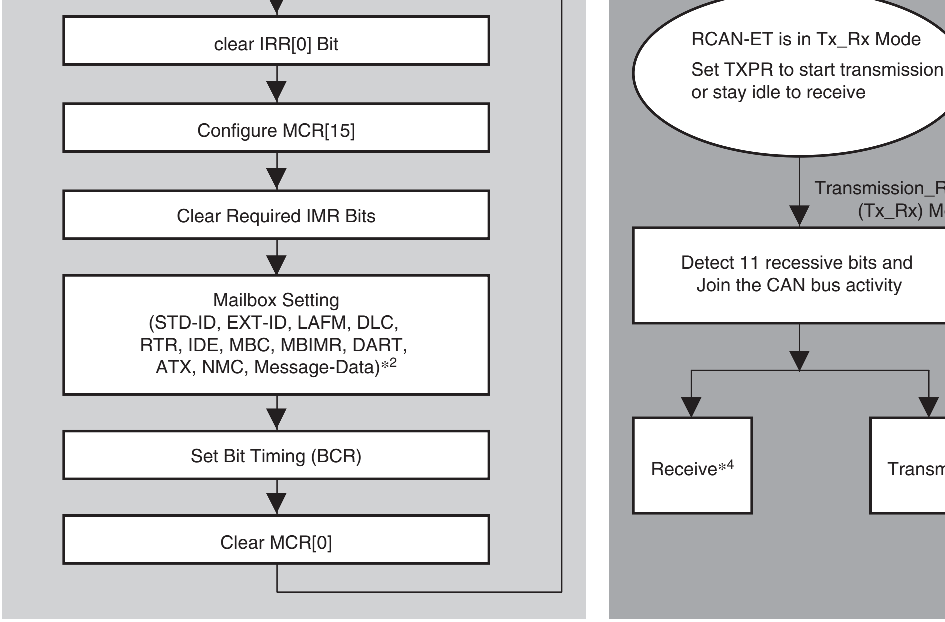
When message error occurs, IRR13 is set in all test modes.

### 19.4.2 Configuration of RCAN-ET

RCAN-ET is considered in configuration mode or after a H/W (Power On Reset)/ S/W (
reset or when in Halt mode. In both conditions RCAN-ET cannot join the CAN Bus acti
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
  reset. After reset, all the registers are initialized, therefore, RCAN-ET needs to be co
  before joining the CAN bus activity. Please read the notes carefully.

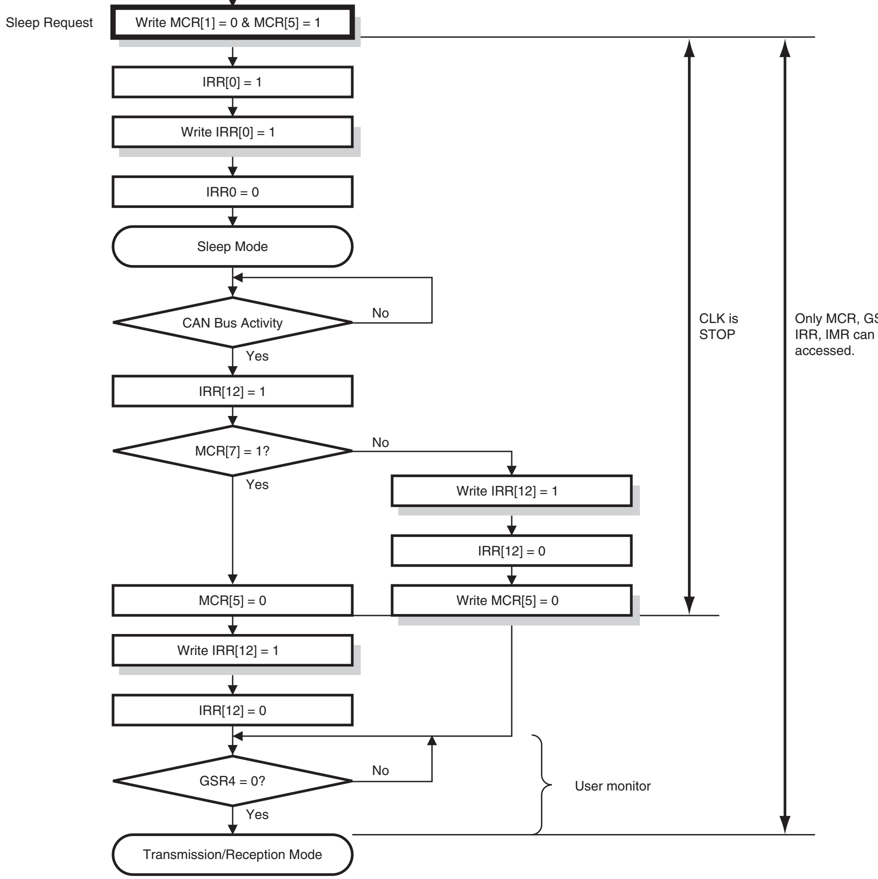


Notes: 1. SW reset could be performed at any time by setting MCR[0] = 1.
2. Mailboxes are comprised of RAMs, therefore, please initialise all the mailboxes enabled by MBC.
3. It takes about one bit time quantum for the value to reach 0.
4. If there is no TXPR set, RCAN-ET will receive the next incoming message.
If there is a TXPR(s) set, RCAN-ET will start transmission of the message and will be arbitrated by the
If it loses the arbitration, it will become a receiver.

**Figure 19.8 Reset Sequence**

When RCAN-ET is in sleep mode the clock for the main blocks of the IP is stopped reduce power consumption. Only the following user registers are clocked and can be MCR, GSR, IRR and IMR. Interrupt related to transmission (TXACK and ABACK) reception (RXPR and RFPR) cannot be cleared when in sleep mode (as TXACK, AI 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 mod


Sleep Request
Write MCR[1] = 0 & MCR[5] = 1
IRR[0] = 1
Write IRR[0] = 1
IRR0 = 0
Sleep Mode
CAN Bus Activity
No
Yes
IRR[12] = 1
MCR[7] = 1?
No
Yes
Write IRR[12] = 1
IRR[12] = 0
MCR[5] = 0
Write MCR[5] = 0
Write IRR[12] = 1
IRR[12] = 0
GSR4 = 0?
No
Yes
Transmission/Reception Mode
User monitor
CLK is
STOP
Only MCR, GS
IRR, IMR can
accessed.

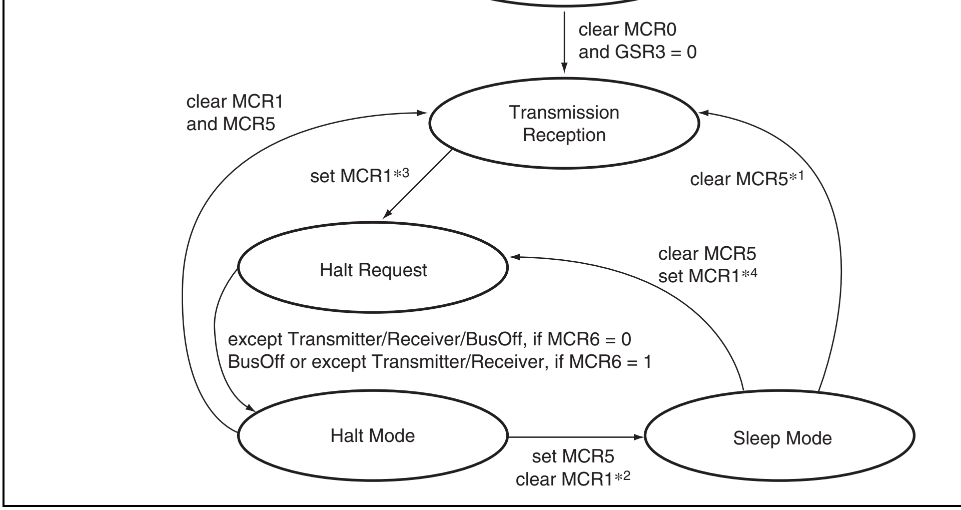


**Figure 19.9 Halt Mode / Sleep Mode**

Notes: 1. MCR5 can be cleared by automatically by detecting a dominant bit on the C
MCR7 is set or by writing "0"

2. MCR1 is cleared in SW. Clearing MCR1 and setting MCR5 have to be carrie
the same instruction.

3. MCR1 must not be cleared in SW, before GSR4 is set. MCR1 can be set auto
in HW when RCAN-ET moves to Bus Off and MCR14 and MCR6 are both

4. When MCR5 is cleared and MCR1 is set at the same time, RCAN-ET moves
Request. Right after that, it moves to Halt Mode with no reception/transmiss

The following table shows conditions to access registers.

2. When TXPR is not set.

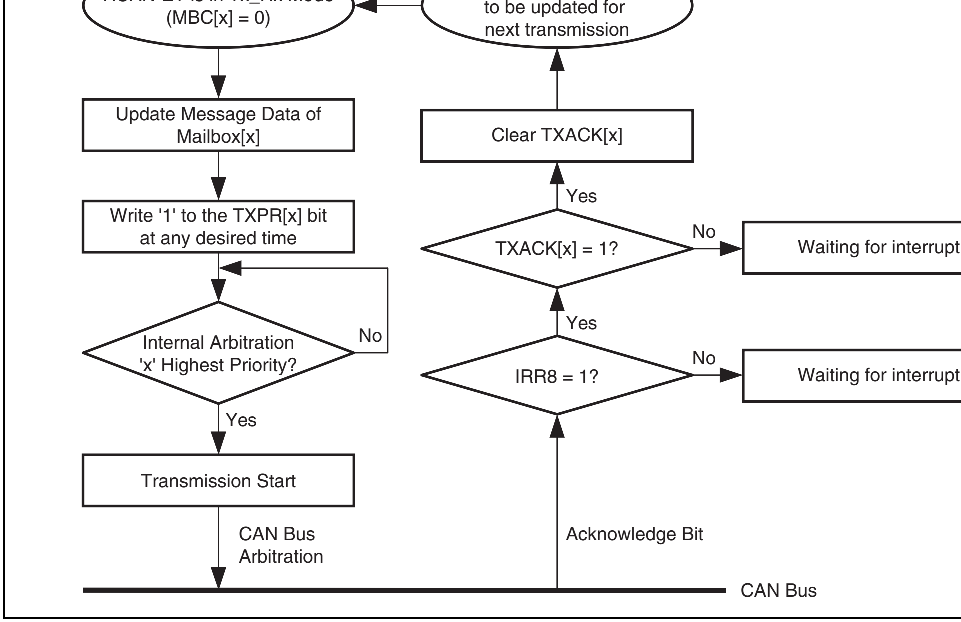


**Figure 19.10 Transmission Request**

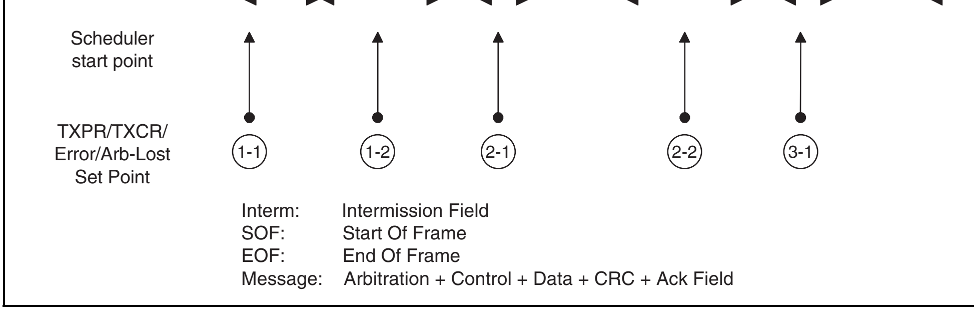

Scheduler start point
TXPR/TXCR/ Error/Arb-Lost Set Point
1-1
1-2
2-1
2-2
3-1
Interm: Intermission Field
SOF: Start Of Frame
EOF: End Of Frame
Message: Arbitration + Control + Data + CRC + Ack Field

**Figure 19.11 Internal Arbitration for Transmission**

The RCAN-ET has two state machines. One is for transmission, and the other is for recep

1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts ru immediately and the transmission is started.

1-2: Operations for both transmission and reception starts at SOF. Since there is no recep frame, RCAN-ET becomes transmitter.

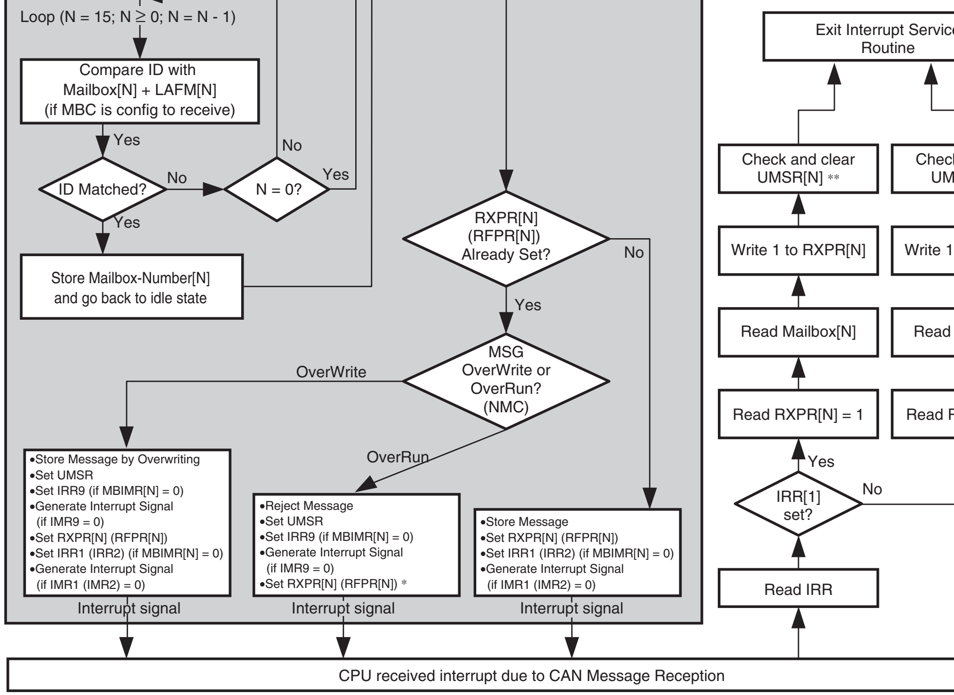
2-1: At crc delimiter, internal arbitration to search next message transmitted starts.

2-2: Operations for both transmission and reception starts at SOF. Because of a reception with higher priority, RCAN-ET becomes receiver. Therefore, Reception is carried ou 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 fr higher priority than reception one, RCAN-ET becomes transmitter.

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


Loop (N = 15; N ≥ 0; N = N - 1)
Compare ID with Mailbox[N] + LAFM[N] (if MBC is config to receive)
Yes
ID Matched?
No
N = 0?
No
Yes
Yes
Store Mailbox-Number[N] and go back to idle state
RXPR[N] (RFPR[N]) Already Set?
No
Yes
MSG OverWrite or OverRun? (NMC)
OverWrite
OverRun
•Store Message by Overwriting
•Set UMSR
•Set IRR9 (if MBIMR[N] = 0)
•Generate Interrupt Signal (if IMR9 = 0)
•Set RXPR[N] (RFPR[N])
•Set IRR1 (IRR2) (if MBIMR[N] = 0)
•Generate Interrupt Signal (if IMR1 (IMR2) = 0)
•Reject Message
•Set UMSR
•Set IRR9 (if MBIMR[N] = 0)
•Generate Interrupt Signal (if IMR9 = 0)
•Set RXPR[N] (RFPR[N]) *
•Store Message
•Set RXPR[N] (RFPR[N])
•Set IRR1 (IRR2) (if MBIMR[N] = 0)
•Generate Interrupt Signal (if IMR1 (IMR2) = 0)
Interrupt signal
Interrupt signal
Interrupt signal
CPU received interrupt due to CAN Message Reception
Exit Interrupt Servic
Routine
Check and clear UMSR[N] **
Checl
UM
Write 1 to RXPR[N]
Write 1
Read Mailbox[N]
Read
Read RXPR[N] = 1
Read F
Yes
IRR[1] set?
No
Read IRR

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 eve 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, UMS and the full Interrupt Service Routine started again. In case of overrun configuration (NMC = 0) is used clear again RXPR[N]/R UMSR[N] when UMSR[N] = 1 and consider the message obsolate.

**Figure 19.12 Message Receive Sequence**

CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM
This also implies that, if the identifier of a received message matches to ID + LAFM of
Mailboxes, the higher numbered Mailbox will always store the relevant messages and th
numbered Mailbox will never receive messages. Therefore, the settings of the identifiers
LAFMs need to be carefully selected.

With regards to the reception of data and remote frames described in the above flow dia
clearing of the UMSR flag after the reading of IRR is to detect situations where a messa
overwritten by a new incoming message stored in the same mailbox while the interrupt s
routine is running. If during the final check of UMSR a overwrite condition is detected t
message needs to be discarded and read again.

In case UMSR is set and the Mailbox is configured for overrun (NMC = 0) the message
valid, however it is obsolete as it is not reflecting the latest message monitored on the C
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
remote frame request interrupt (IRR2) and data frame received interrupt (IRR1) and also
Receive Flags (RXPR and RFPR) are set. In an analogous way, the overwriting of a data
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 asser
a Data Frame is initially received, then RXPR and IRR1 are both asserted. If a Remote F
then received before the Data Frame has been read, then RFPR and IRR2 are NOT set. I
UMSR of the corresponding Mailbox will still be set.

Confirm that the corresponding TXPR is not set. The configuration can be change
Halt or reset state. Please note that it might take longer for RCAN-ET to transit to
if it is receiving or transmitting a message (as the transition to the halt state is dela
the end of the reception/transmission), and also RCAN-ET will not be able to
receive/transmit messages during the Halt state.

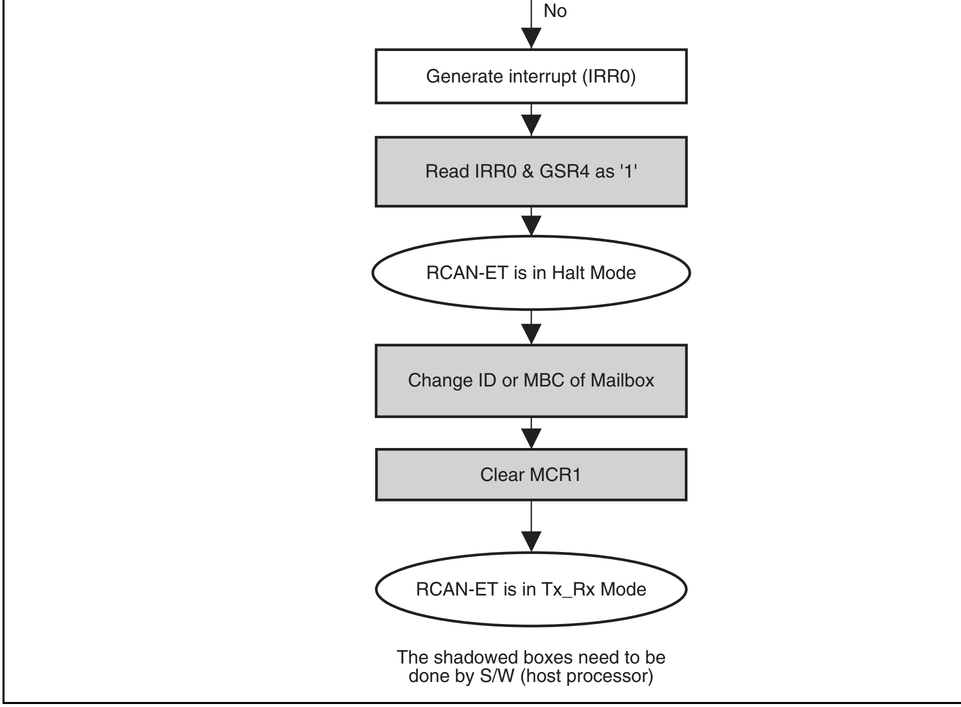
In case RCAN-ET is in the Bus Off state the transition to halt state depends on the
configuration of the bit 6 of MCR and also bit and 14 of MCR.

- Change configuration (ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART, MBC
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 RC
is a receiver. RCAN-ET will be moving into Halt Mode after completing the current r
Please note that it might take longer if RCAN-ET is receiving or transmitting a messa
transition to the halt state is delayed until the end of the reception/transmission), and a
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.



The shadowed boxes need to be done by S/W (host processor)

**Figure 19.13 Change ID of Receive Box or Change Receive Box to Transmit**

| | | | | |
|---|---|---|---|---|
| RCAN-ET_0 | ERS_0 | Error Passive Mode (TEC ≥ 128 or REC ≥ 128) | IRR5 | Not |
| | | 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] | Pos |
| | RM1_0*[2] | Remote frame reception | IRR2*[3] | |
| | SLE_0 | Message transmission/transmission disabled (slot empty) | IRR8 | Not |

Notes: 1. Available only in Test Mode.

2. RM0_0 is an interrupt generated by the remote request pending flag for mailbo (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1_0 is interrupt generated by the remote request pending flag for mailbox n (RFPR0[r 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 frame request interrupt flag for mailboxes 0 to 15.

4. The DTC can be activated only by the RM0_0 interrupt.

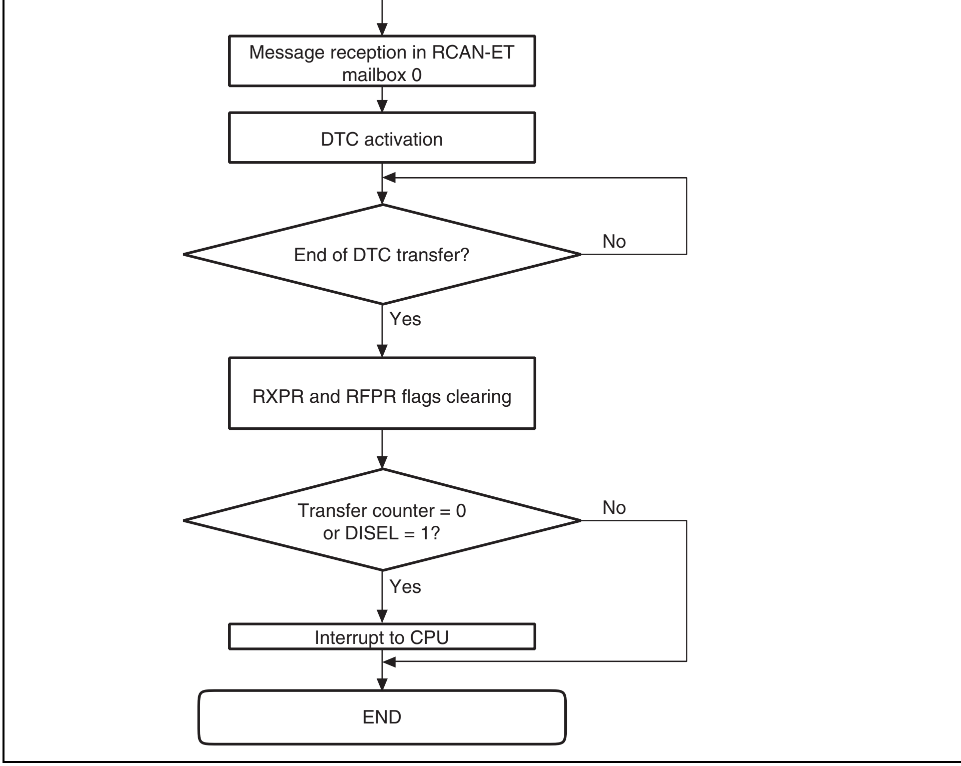
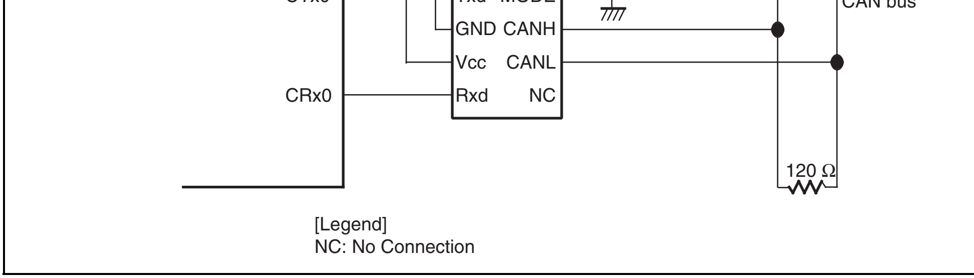

Message reception in RCAN-ET mailbox 0
DTC activation
End of DTC transfer?
No
Yes
RXPR and RFPR flags clearing
Transfer counter = 0 or DISEL = 1?
No
Yes
Interrupt to CPU
END

**Figure 19.14 DTC Transfer Flowchart**



[Legend]
NC: No Connection

**Figure 19.15 High-Speed CAN Interface Using HA13721**

- Hardware reset

  RCAN-ET is reset to the initial state by power-on reset or on entering module stop m software standby mode.

- Software reset

  By setting the MCR0 bit in Master Control Register (MCR), RCAN-ET registers, ex 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 re 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 initiali reset because they are in RAM. After power-on reset, all mailboxes should be initialized 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. There not make access in CAN sleep mode except for access to the MCR, GSR, IRR, and IMR

### 19.8.4 Register Access

If the mailbox area is accessed while the CAN communication circuitry in RCAN-ET is received CAN bus frame in a mailbox, a 0 to five peripheral clock cycles of wait state is generated.

| Port | (Related Module) | (Related Module) | (Related Module) | (Related Mo |
|---|---|---|---|---|
| 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 (P |
| | 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 (S |
| | PA9 I/O (port) | TCLKD input (MTU2) | $\overline{\text{POE8}}$ input (POE) | TXD2 output ( |
| | 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.

| | | | | |
|---|---|---|---|---|
| | | (UBC)* | | |
| 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 inpu |
| PA9 I/O (port) | $\overline{\text{WAIT}}$ input (BSC) | TCLKD input (MTU2) | $\overline{\text{POE8}}$ input (POE) | TXD2 outp |
| 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) | Functio (Relatec |
|---|---|---|---|---|---|
| B | PB2 I/O (port) | IRQ0 input (INTC) | $\overline{\text{POE0}}$ input (POE) | TIC5VS input (MTU2S) | SCL I/O (I |
| | PB3 I/O (port) | IRQ1 input (INTC) | $\overline{\text{POE1}}$ input (POE) | TIC5V input (MTU2) | SDA I/O ( |
| | 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) | — | — | — |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | (POE) | (MTU2) | |
| | 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 Mo |
|---|---|---|---|---|
| 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) | — | — |

| | | | |
|---|---|---|---|
| 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 ( |
| 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.

| | | | | |
|---|---|---|---|---|
| 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)* | $\overline{\text{ASEBRK}}$ (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.

| | |
|---|---|
| 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.

| | |
|---|---|
| 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.

|  |  |  |
|---|---|---|
| 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/$\overline{\text{ADTRG}}$ |

| | | |
|---|---|---|
| 32 | PB4 | PB4/IRQ2/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 |

| | | |
|---|---|---|
| 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 t
($\overline{\text{ASEMD0}}$ = low).

2. Function enabled on the SH7136 only.

| | | |
|---|---|---|
| 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 |

| | | |
|---|---|---|
| 47 | PB1 | PB1/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 |

| | | |
|---|---|---|
| 21 | A13 | PE6/A13/TIOC2A/SCK1 |
| 20 | A14 | PE7/A14/TIOC2B |
| 19 | A15 | PE8/A15/TIOC3A |
| 17 | PE9 | PE9/TIOC3B |
| 18 | $\overline{\text{CS0}}$ | PE10/$\overline{\text{CS0}}$/TIOC3C |
| 15 | PE11 | PE11/TIOC3D |
| 13 | PE12 | PE12/TIOC4A |
| 12 | PE13 | PE13/TIOC4B/$\overline{\text{MRES}}$ |
| 10 | PE14 | PE14/TIOC4C |
| 9 | PE15 | PE15/TIOC4D/$\overline{\text{IRQOUT}}$ |
| 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 |
| 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 |

| | | |
|---|---|---|
| 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 t
($\overline{\text{ASEMD0}}$ = low).

2. Function enabled on the SH7137 only.

|  |  |  |  |  |
|---|---|---|---|---|
| 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 | $\overline{\text{RES}}$ | $\overline{\text{RES}}$ | $\overline{\text{RES}}$ | $\overline{\text{RES}}$ |
| 100 | $\overline{\text{WDTOVF}}$ | $\overline{\text{WDTOVF}}$ | $\overline{\text{WDTOVF}}$ | $\overline{\text{WDTOVF}}$ |
| 73 | NMI | NMI | NMI | NMI |
| 76 | $\overline{\text{ASEMD0}}$ | $\overline{\text{ASEMD0}}$ | $\overline{\text{ASEMD0}}$ | $\overline{\text{ASEMD0}}$ |
| 69 | PA0 | PA0/A0/$\overline{\text{POE0}}$/RXD0 | PA0 | PA0/$\overline{\text{POE0}}$/RXD0 |
| 68 | PA1 | PA1/A1/$\overline{\text{POE1}}$/TXD0 | PA1 | PA1/$\overline{\text{POE1}}$/TXD0 |
| 67 | PA2 | PA2/A2/IRQ0/$\overline{\text{POE2}}$/SCK0 | PA2 | PA2/IRQ0/$\overline{\text{POE2}}$/S |
| 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/$\overline{\text{RD}}$/$\overline{\text{UBCTRG}}$*2/TCLKA/ $\overline{\text{POE4}}$ | PA6 | PA6/$\overline{\text{UBCTRG}}$*2/T |
| 61 | PA7 | PA7/TCLKB/$\overline{\text{POE5}}$/SCK2 | PA7 | PA7/TCLKB/$\overline{\text{POE5}}$ |
| 60 | PA8 | PA8/$\overline{\text{WRL}}$/TCLKC/$\overline{\text{POE6}}$/ RXD2 | PA8 | PA8/TCLKC/$\overline{\text{POE}}$ |

| | | | | |
|---|---|---|---|---|
| 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/$\overline{\text{POE0}}$/TI |
| 45 | PB3 | PB3/A17/IRQ1/$\overline{\text{POE1}}$/TIC5V/SDA | PB3 | PB3/IRQ1/$\overline{\text{POE1}}$/TI |
| 44 | PB4 | PB4/A18/IRQ2/$\overline{\text{POE4}}$/TIC5US | PB4 | PB4/IRQ2/$\overline{\text{POE4}}$/TI |
| 43 | PB5 | PB5/A19/IRQ3/$\overline{\text{POE5}}$/TIC5U | PB5 | PB5/IRQ3/$\overline{\text{POE5}}$/TI |
| 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 |

| | | | | |
|---|---|---|---|---|
| 21 | PE6 | PE6/A13/TIOC2A/SCK1 | PE6 | PE6/TIOC2A/SCK |
| 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/$\overline{\text{MR}}$ |
| 10 | PE14 | PE14/TIOC4C | PE14 | PE14/TIOC4C |
| 9 | PE15 | PE15/TIOC4D/$\overline{\text{IRQOUT}}$ | PE15 | PE15/TIOC4D/$\overline{\text{IRC}}$ |
| 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 |

|  |  |  |  |  |
|---|---|---|---|---|
| 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 t ($\overline{\text{ASEMD0}}$ = low).

2. Function enabled on the SH7137 only.

| | | | | | |
|---|---|---|---|---|---|
| Port A control register L2 | PACRL2 | R/W | H'0000* | H'FFFFD114 | 8, 1 |
| Port A control register L1 | PACRL1 | R/W | H'0000* | H'FFFFD116 | 8, 1 |
| Port B I/O register L | PBIORL | R/W | H'0000 | H'FFFFD186 | 8, 1 |
| Port B control register L2 | PBCRL2 | R/W | H'0000 | H'FFFFD194 | 8, 1 |
| Port B control register L1 | PBCRL1 | R/W | H'0000* | H'FFFFD196 | 8, 1 |
| Port D I/O register L | PDIORL | R/W | H'0000 | H'FFFFD286 | 8, 1 |
| Port D control register L3 | PDCRL3 | R/W | H'0000 | H'FFFFD292 | 8, 1 |
| Port D control register L2 | PDCRL2 | R/W | H'0000* | H'FFFFD294 | 8, 1 |
| Port D control register L1 | PDCRL1 | R/W | H'0000* | H'FFFFD296 | 8, 1 |
| Port E I/O register H | PEIORH | R/W | H'0000 | H'FFFFD304 | 8, 1 |
| Port E I/O register L | PEIORL | R/W | H'0000 | H'FFFFD306 | 8, 1 |
| Port E control register H2 | PECRH2 | R/W | H'0000 | H'FFFFD30C | 8, 1 |
| Port E control register H1 | PECRH1 | R/W | H'0000 | H'FFFFD30E | 8, 1 |
| Port E control register L4 | PECRL4 | R/W | H'0000 | H'FFFFD310 | 8, 1 |
| Port E control register L3 | PECRL3 | R/W | H'0000* | H'FFFFD312 | 8, 1 |
| Port E control register L2 | PECRL2 | R/W | H'0000* | H'FFFFD314 | 8, 1 |
| Port E control register L1 | PECRL1 | R/W | H'0000 | H'FFFFD316 | 8, 1 |
| IRQOUT function control register | IFCR | R/W | H'0000 | H'FFFFD322 | 8, 1 |

Note: For SH7132 and SH7137, the initial value differs in the on-chip ROM enabled/disa external-extension mode. For details, refer to register descriptions in this section.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
| | PA15<br>IOR | PA14<br>IOR | PA13<br>IOR | PA12<br>IOR | PA11<br>IOR | PA10<br>IOR | PA9<br>IOR | PA8<br>IOR | PA7<br>IOR | PA6<br>IOR | PA5<br>IOR | PA4<br>IOR | PA3<br>IOR | PA2<br>IOR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | 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 fun
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 |
| | - | PA15<br>MD2 | PA15<br>MD1 | PA15<br>MD0 | - | PA14<br>MD2 | PA14<br>MD1 | PA14<br>MD0 | - | PA13<br>MD2 | PA13<br>MD1 | PA13<br>MD0 | - | PA12<br>MD2 |
| Initial value: | 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 |

| | | | | |
|---|---|---|---|---|
| | | | | Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value always be 0. |
| 10 | PA14MD2 | 0 | R/W | PA14 Mode |
| 9 | PA14MD1 | 0 | R/W | Select the function of the PA14/RXD1/SSI p |
| 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 always be 0. |
| 6 | PA13MD2 | 0 | R/W | PA13 Mode |
| 5 | PA13MD1 | 0 | R/W | Select the function of the PA13/SCK1/SSCK |
| 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 always be 0. |

| | | | | |
|---|---|---|---|---|
| 14 | PA11MD2 | 0 | R/W | PA11 Mode |
| 13 | PA11MD1 | 0 | R/W | Select the function of the PA11/TXD0/$\overline{\text{ADTR}}$ |
| 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 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 always be 0. |
| 6 | PA9MD2 | 0 | R/W | PA9 Mode |
| 5 | PA9MD1 | 0 | R/W | Select the function of the PA9/TCLKD/TXD2 |
| 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 |

110: RXD2 input (SCI)

111: $\overline{\text{POE6}}$ input (POE)

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA7 MD2 | PA7 MD1 | PA7 MD0 | - | PA6 MD2 | PA6 MD1 | PA6 MD0 | - | PA5 MD2 | PA5 MD1 | PA5 MD0 | - | PA4 MD2 |
| Initial value: | 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s 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/$\overline{\text{POE5}}$/ 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 s always be 0. |

| | | | | |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value<br>always be 0. |
| 6 | PA5MD2 | 0 | R/W | PA5 Mode |
| 5 | PA5MD1 | 0 | R/W | Select the function of the PA5/IRQ3/SCK1 p |
| 4 | PA5MD0 | 0 | R/W | 000: PA5 I/O (port)<br>001: SCK1 I/O (SCI)<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<br>always be 0. |
| 2 | PA4MD2 | 0 | R/W | PA4 Mode |
| 1 | PA4MD1 | 0 | R/W | Select the function of the PA4/IRQ2/TXD1 p |
| 0 | PA4MD0 | 0 | R/W | 000: PA4 I/O (port)<br>001: TXD1 output (SCI)<br>111: IRQ2 input (INTC)<br>Other than above: Setting prohibited |

Note: * Function enabled on the SH7136 only. Do not use this setting on the SH7131

|  |  |  |  |  |
|---|---|---|---|---|
| 14 | PA3MD2 | 0 | R/W | PA3 Mode |
| 13 | PA3MD1 | 0 | R/W | Select the function of the PA3/IRQ1/RXD1 pi |
| 12 | PA3MD0 | 0 | R/W | 000: PA3 I/O (port) |
|  |  |  |  | 001: RXD1 input (SCI) |
|  |  |  |  | 111: IRQ1 input (INTC) |
|  |  |  |  | Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved |
|  |  |  |  | This bit is always read as 0. The write value s always be 0. |
| 10 | PA2MD2 | 0 | R/W | PA2 Mode |
| 9 | PA2MD1 | 0 | R/W | Select the function of the PA2/IRQ0/$\overline{\text{POE2}}$/S |
| 8 | PA2MD0 | 0 | R/W | 000: PA2 I/O (port) |
|  |  |  |  | 001: SCK0 I/O (SCI) |
|  |  |  |  | 011: IRQ0 input (INTC) |
|  |  |  |  | 111: $\overline{\text{POE2}}$ input (POE) |
|  |  |  |  | Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved |
|  |  |  |  | This bit is always read as 0. The write value s 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 p |
| 4 | PA1MD0 | 0 | R/W | 000: PA1 I/O (port) |
|  |  |  |  | 001: TXD0 output (SCI) |
|  |  |  |  | 111: $\overline{\text{POE1}}$ input (POE) |
|  |  |  |  | Other than above: Setting prohibited |

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PA15<br>MD2 | PA15<br>MD1 | PA15<br>MD0 | - | PA14<br>MD2 | PA14<br>MD1 | PA14<br>MD0 | - | PA13<br>MD2 | PA13<br>MD1 | PA13<br>MD0 | - | PA12<br>MD2 |
| Initial value: | 0 | 0 | 0 | 0*1 | 0 | 0*2 | 0 | 0 | 0 | 0*2 | 0 | 0 | 0 | 0*2 |
| 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 |

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<br>always be 0. |
| 14 | PA15MD2 | 0 | R/W | PA15 Mode |
| 13 | PA15MD1 | 0 | R/W | Select the function of the PA15/CK/TXD1/S |
| 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<br>always be 0. |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | This bit is always read as 0. The write value s always be 0. |
| 6 | PA13MD2 | 0*[2] | R/W | PA13 Mode |
| 5 | PA13MD1 | 0 | R/W | Select the function of the PA13/A9/SCK1/SS |
| 4 | PA13MD0 | 0 | R/W | 000: PA13 I/O (port) |
|  |  |  |  | 100: A9 output (BSC)*[3] |
|  |  |  |  | 101: SSCK I/O (SSU) |
|  |  |  |  | 110: SCK1 I/O (SCI) |
|  |  |  |  | Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved |
|  |  |  |  | This bit is always read as 0. The write value s always be 0. |
| 2 | PA12MD2 | 0*[2] | R/W | PA12 Mode |
| 1 | PA12MD1 | 0 | R/W | Select the function of the PA12/A8/SCK0/$\overline{\text{SC}}$ |
| 0 | PA12MD0 | 0 | R/W | 000: PA12 I/O (port) |
|  |  |  |  | 100: A8 output (BSC)*[3] |
|  |  |  |  | 101: $\overline{\text{SCS}}$ I/O (SSU) |
|  |  |  |  | 110: SCK0 I/O (SCI) |
|  |  |  |  | Other than above: Setting prohibited |

Notes: 1. The initial value is 1 in the on-chip ROM enabled/disabled external-extension r

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-e mode. Do not set to this value in single-chip mode.

| | | | | |
|---|---|---|---|---|
| | | | | always be 0. |
| 14 | PA11MD2 | 0*[1] | R/W | PA11 Mode |
| 13 | PA11MD1 | 0 | R/W | Select the function of the PA11/A7/TXD0/$\overline{\text{AI}}$ |
| 12 | PA11MD0 | 0 | R/W | 000: PA11 I/O (port) |
| | | | | 010: $\overline{\text{ADTRG}}$ input (A/D) |
| | | | | 100: A7 output (BSC)*[2] |
| | | | | 110: TXD0 output (SCI) |
| | | | | Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved |
| | | | | This bit is always read as 0. The write value always be 0. |
| 10 | PA10MD2 | 0*[1] | R/W | PA10 Mode |
| 9 | PA10MD1 | 0 | R/W | Select the function of the PA10/A6/RXD0 pi |
| 8 | PA10MD0 | 0 | R/W | 000: PA10 I/O (port) |
| | | | | 100: A6 output (BSC)*[2] |
| | | | | 110: RXD0 input (SCI) |
| | | | | Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved |
| | | | | This bit is always read as 0. The write value always be 0. |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s always be 0. |
| 2<br>1<br>0 | PA8MD2<br>PA8MD1<br>PA8MD0 | 0*[1]<br>0<br>0 | R/W<br>R/W<br>R/W | PA8 Mode<br>Select the function of the PA8/$\overline{\text{WRL}}$/TCLKC/$\overline{\text{POE6}}$/RXD2 pin.<br>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-ex mode. Do not set to this value in single-chip mode.

| | | | | |
|---|---|---|---|---|
| | | | | 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/$\overline{\text{POE5}}$ 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 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 |

| | | | | |
|---|---|---|---|---|
| | | | | 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 s 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 |
| 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-ex 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.

| | | | | |
|---|---|---|---|---|
| | | | | always be 0. |
| 14 | PA3MD2 | $0^{*1}$ | R/W | PA3 Mode |
| 13 | PA3MD1 | 0 | R/W | Select the function of the PA3/A3/IRQ1/RXD |
| 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 always be 0. |
| 10 | PA2MD2 | $0^{*1}$ | R/W | PA2 Mode |
| 9 | PA2MD1 | 0 | R/W | Select the function of the PA2/A2/IRQ0/$\overline{\text{PO}}$ |
| 8 | PA2MD0 | 0 | R/W | pin.<br>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 always be 0. |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | This bit is always read as 0. The write value s always be 0. |
| 2 | PA0MD2 | 0*[1] | R/W | PA0 Mode |
| 1 | PA0MD1 | 0 | R/W | Select the function of the PA0/A0/$\overline{\text{POE0}}$/RXD |
| 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-e mode. Do not set to this value in single-chip mode.

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 sh 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | PB7 IOR | PB6 IOR | PB5 IOR | PB4 IOR | PB3 IOR | PB2 IOR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R/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 f 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PB7 MD2 | PB7 MD1 | PB7 MD0 | - | PB6 MD2 | PB6 MD1 | PB6 MD0 | - | PB5 MD2 | PB5 MD1 | PB5 MD0 | - | PB4 MD2 |
| Initial value: | 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 |

|  |  |  |  |  |
|---|---|---|---|---|
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s 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 s always be 0. |
| 6 | PB5MD2 | 0 | R/W | PB5 Mode |
| 5 | PB5MD1 | 0 | R/W | Select the function of the PB5/IRQ3/$\overline{\text{POE5}}$/T |
| 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 s always be 0. |

- Port B Control Register L1 (PBCRL1)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | - | PB3<br>MD2 | PB3<br>MD1 | PB3<br>MD0 | - | PB2<br>MD2 | PB2<br>MD1 | PB2<br>MD0 | - | - | - | - | - | - |
| Initial value: | 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value always be 0. |
| 14<br>13<br>12 | PB3MD2<br>PB3MD1<br>PB3MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PB3 Mode<br>Select the function of the PB3/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>Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | Other than above: Setting prohibited |
| 7 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va<br>should always be 0. |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | This bit is always read as 0. The write value<br>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 pi |
| 12 | PB7MD0 | 0 | R/W | 000: PB7 I/O (port) |
|  |  |  |  | 101: $\overline{\text{CS1}}$ output (BSC)* |
|  |  |  |  | 110: CRx0 input (RCAN-ET) |
|  |  |  |  | Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved |
|  |  |  |  | This bit is always read as 0. The write value<br>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 p |
| 8 | PB6MD0 | 0 | R/W | 000: PB6 I/O (port) |
|  |  |  |  | 101: $\overline{\text{WAIT}}$ input (BSC)* |
|  |  |  |  | 110: CTx0 output (RCAN-ET) |
|  |  |  |  | Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved |
|  |  |  |  | This bit is always read as 0. The write value<br>always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s always be 0. |
| 2<br>1<br>0 | PB4MD2<br>PB4MD1<br>PB4MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PB4 Mode<br>Select the function of the PB4/A18/IRQ2/$\overline{\text{POI}}$ TIC5US pin.<br>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-ex mode. Do not set to this value in single-chip mode.

| | | | | |
|---|---|---|---|---|
| | | | | 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/$\overline{\text{PO}}$ 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 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/$\overline{\text{PO}}$ 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 |

| | | | | |
|---|---|---|---|---|
| | | | | Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s 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 |
| 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-ex mode. Do not set to this value in single-chip mode.

Bits 15 to 11 of PDIORL are reserved. These bits are always read as 0. The write value s 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | PD10 IOR | PD9 IOR | PD8 IOR | PD7 IOR | PD6 IOR | PD5 IOR | PD4 IOR | PD3 IOR | PD2 IOR |
| Initial value: | 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 | R | R | R | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | 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 va 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 s always be 0. |

| | | | | |
|---|---|---|---|---|
| 2 | PD8MD2 | 0 | R/W | PD8 Mode |
| 1 | PD8MD1 | 0 | R/W | Select the function of the PD8/SCK2/SSCK |
| 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 |

| | | | | |
|---|---|---|---|---|
| | | | | 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 |
| 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 s 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 s always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | 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-
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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PD3 MD2 | PD3 MD1 | PD3 MD0 | - | PD2 MD2 | PD2 MD1 | PD2 MD0 | - | PD1 MD2 | PD1 MD1 | PD1 MD0 | - | PD0 MD2 |
| Initial value: | 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 |

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<br>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) |
| | | | | 001: D2 I/O (BSC)*[2] |
| | | | | 110: SCK0 I/O (SCI) |
| | | | | Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved |
| | | | | This bit is always read as 0. The write value s 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) |
| | | | | 001: D1 I/O (BSC)*[2] |
| | | | | 110: TXD0 output (SCI) |
| | | | | Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved |
| | | | | This bit is always read as 0. The write value s 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) |
| | | | | 001: D0 I/O (BSC)*[2] |
| | | | | 110: RXD0 input (SCI) |
| | | | | 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-ex mode. Do not set to this value in single-chip mode.

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 sh
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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | PE21 IOR | PE20 IOR | PE19 IOR | PE18 IOR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | MD1 | MD0 | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W: | R | R | R | R | R | R | R | R | R | R | R/W | R/W | 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 va<br>should always be 0. |
| 5 | PE21MD1 | 0 | R/W | PE21 Mode |
| 4 | PE21MD0 | 0 | R/W | Select the function of the PE21/TIOC4DS/$\overline{\text{TR}}$<br>Fixed to $\overline{\text{TRST}}$ input when using the E10A ($\overline{\text{A}}$<br>= 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 va<br>should always be 0. |
| 1 | PE20MD1 | 0 | R/W | PE20 Mode |
| 0 | PE20MD0 | 0 | R/W | Select the function of the PE20/TIOC4CS/TM<br>Fixed to TMS input when using the E10A ($\overline{\text{AS}}$<br>low).<br>00: PE20 I/O (port)<br>01: TIOC4CS I/O (MTU2S)<br>Other than above: Setting prohibited |

| | | | | |
|---|---|---|---|---|
| 13 | PE19MD1 | 0 | R/W | PE19 Mode |
| 12 | PE19MD0 | 0 | R/W | Select the function of the PE19/TIOC4BS/T<br>Fixed to TDO output when using the E10A (<br>= 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 v<br>should always be 0. |
| 9 | PE18MD1 | 0 | R/W | PE18 Mode |
| 8 | PE18MD0 | 0 | R/W | Select the function of the PE18/TIOC4AS/T<br>Fixed to TDI input when using the E10A ($\overline{\text{AS}}$<br>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 v<br>should always be 0. |

This bit is always read as 0. The write value
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. F $\overline{\text{ASEBRKAK}}$ output/$\overline{\text{ASEBRK}}$ input when usir 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE15 MD2 | PE15 MD1 | PE15 MD0 | - | PE14 MD2 | PE14 MD1 | PE14 MD0 | - | - | PE13 MD1 | PE13 MD0 | - | PE12 MD2 |
| Initial value: | 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | 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) |
| | | | | 001: TIOC4C I/O (MTU2) |
| | | | | Other than above: Setting prohibited |
| 7, 6 | — | All 0 | R | Reserved |
| | | | | These bits are always read as 0. The write v should always be 0. |
| 5 | PE13MD1 | 0 | R/W | PE13 Mode |
| 4 | PE13MD0 | 0 | R/W | Select the function of the PE13/TIOC4B/$\overline{\text{MR}}$ |
| | | | | 00: PE13 I/O (port) |
| | | | | 01: TIOC4B I/O (MTU2) |
| | | | | 10: $\overline{\text{MRES}}$ input (INTC) |
| | | | | Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved |
| | | | | This bit is always read as 0. The write value 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) |
| | | | | 001: TIOC4A I/O (MTU2) |
| | | | | Other than above: Setting prohibited |

|  |  |  |  |  |
|---|---|---|---|---|
| 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) |
|  |  |  |  | 001: TIOC3D I/O (MTU2) |
|  |  |  |  | Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s 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) |
|  |  |  |  | 001: TIOC3C I/O (MTU2) |
|  |  |  |  | Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s 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) |
|  |  |  |  | 001: TIOC3B I/O (MTU2) |
|  |  |  |  | Other than above: Setting prohibited |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s always be 0. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | - | PE7<br>MD2 | PE7<br>MD1 | PE7<br>MD0 | - | PE6<br>MD2 | PE6<br>MD1 | PE6<br>MD0 | - | PE5<br>MD2 | PE5<br>MD1 | PE5<br>MD0 | - | PE4<br>MD2 |
| Initial value: | 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value<br>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<br>always be 0. |
| 10 | PE6MD2 | 0 | R/W | PE6 Mode |
| 9 | PE6MD1 | 0 | R/W | Select the function of the PE6/TIOC2A/SCK |
| 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<br>always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| | | | | always be 0. |
| 2<br>1<br>0 | PE4MD2<br>PE4MD1<br>PE4MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PE4 Mode<br>Select the function of the PE4/TIOC1A/RXD1<br>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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE3 MD2 | PE3 MD1 | PE3 MD0 | - | PE2 MD2 | PE2 MD1 | PE2 MD0 | - | PE1 MD2 | PE1 MD1 | PE1 MD0 | - | - |
| Initial value: | 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s<br>always be 0. |
| 14<br>13<br>12 | PE3MD2<br>PE3MD1<br>PE3MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PE3 Mode<br>Select the function of the PE3/TIOC0D/SCK0<br>000: PE3 I/O (port)<br>001: TIOC0D I/O (MTU2)<br>110: SCK0 I/O (SCI)<br>Other than above: Setting prohibited |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value always be 0. |
| 6 | PE1MD2 | 0 | R/W | PE1 Mode |
| 5 | PE1MD1 | 0 | R/W | Select the function of the PE1/TIOC0B/RXD |
| 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 v 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 |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  |  | These bits are always read as 0. The write va should always be 0. |
| 5 | PE21MD1 | 0 | R/W | PE21 Mode |
| 4 | PE21MD0 | 0 | R/W | Select the function of the PE21/$\overline{\text{WRL}}$/TIOC4D pin. Fixed to $\overline{\text{TRST}}$ input when using the E10 ($\overline{\text{ASEMD0}}$ = 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 va should always be 0. |
| 1 | PE20MD1 | 0 | R/W | PE20 Mode |
| 0 | PE20MD0 | 0 | R/W | Select the function of the PE20/TIOC4CS/TM Fixed to TMS input when using the E10A ($\overline{\text{AS}}$ 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-e mode. Do not set to this value in single-chip mode.

| | | | | |
|---|---|---|---|---|
| 13 | PE19MD1 | 0 | R/W | PE19 Mode |
| 12 | PE19MD0 | 0 | R/W | Select the function of the PE19/$\overline{\text{RD}}$/TIOC4B pin. Fixed to TDO output when using the E1 ($\overline{\text{ASEMD0}}$ = 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 v should always be 0. |
| 9 | PE18MD1 | 0 | R/W | PE18 Mode |
| 8 | PE18MD0 | 0 | R/W | Select the function of the PE18/$\overline{\text{CS1}}$/TIOC4 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 v should always be 0. |

| | | | | |
|---|---|---|---|---|
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s<br>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/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$<br>Fixed to $\overline{\text{ASEBRKAK}}$ output/$\overline{\text{ASEBRK}}$ input w<br>using the E10A ($\overline{\text{ASEMD0}}$ = low).<br>000: PE16 I/O (port)<br>001: TIOC3BS I/O (MTU2S)<br>010: $\overline{\text{WAIT}}$ input (BSC)*<br>Other than above: Setting prohibited |
| 0 | PE16MD0 | 0 | R/W | |

Note: * This function is available only in the on-chip ROM enabled/disabled external-ex 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE15 MD2 | PE15 MD1 | PE15 MD0 | - | PE14 MD2 | PE14 MD1 | PE14 MD0 | - | - | PE13 MD1 | PE13 MD0 | - | PE12 MD2 |
| Initial value: | 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value s<br>always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | always be 0. |
| 10<br>9<br>8 | PE14MD2<br>PE14MD1<br>PE14MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PE14 Mode<br>Select the function of the PE14/TIOC4C pin<br>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 v<br>should always be 0. |
| 5<br>4 | PE13MD1<br>PE13MD0 | 0<br>0 | R/W<br>R/W | PE13 Mode<br>Select the function of the PE13/TIOC4B/$\overline{\text{MR}}$<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<br>always be 0. |
| 2<br>1<br>0 | PE12MD2<br>PE12MD1<br>PE12MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PE12 Mode<br>Select the function of the PE12/TIOC4A pin<br>000: PE12 I/O (port)<br>001: TIOC4A I/O (MTU2)<br>Other than above: Setting prohibited |

| | | | | |
|---|---|---|---|---|
| | | | | always be 0. |
| 14<br>13<br>12 | PE11MD2<br>PE11MD1<br>PE11MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PE11 Mode<br>Select the function of the PE11/TIOC3D pin.<br>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 s always be 0. |
| 10<br>9<br>8 | PE10MD2<br>PE10MD1<br>PE10MD0 | 0*1<br>0<br>0 | R/W<br>R/W<br>R/W | PE10 Mode<br>Select the function of the PE10/$\overline{\text{CS0}}$/TIOC3C<br>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 s always be 0. |
| 6<br>5<br>4 | PE9MD2<br>PE9MD1<br>PE9MD0 | 0<br>0<br>0 | R/W<br>R/W<br>R/W | PE9 Mode<br>Select the function of the PE9/TIOC3B pin.<br>000: PE9 I/O (port)<br>001: TIOC3B I/O (MTU2)<br>Other than above: Setting prohibited |

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-
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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | PE7<br>MD2 | PE7<br>MD1 | PE7<br>MD0 | - | PE6<br>MD2 | PE6<br>MD1 | PE6<br>MD0 | - | PE5<br>MD2 | PE5<br>MD1 | PE5<br>MD0 | - | PE4<br>MD2 |
| Initial value: | 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 |

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<br>always be 0. |
| 14 | PE7MD2 | 0*1 | R/W | PE7 Mode |
| 13 | PE7MD1 | 0 | R/W | Select the function of the PE7/A14/TIOC2B |
| 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<br>always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | This bit is always read as 0. The write value s<br>always be 0. |
| 6 | PE5MD2 | 0*1 | R/W | PE5 Mode |
| 5 | PE5MD1 | 0 | R/W | Select the function of the PE5/A12/TIOC1B/T |
| 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 s<br>always be 0. |
| 2 | PE4MD2 | 0*1 | R/W | PE4 Mode |
| 1 | PE4MD1 | 0 | R/W | Select the function of the PE4/A11/TIOC1A/F |
| 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-e mode. Do not set to this value in single-chip mode.

| | | | | |
|---|---|---|---|---|
| 14 | PE3MD2 | 0 | R/W | PE3 Mode |
| 13 | PE3MD1 | 0 | R/W | Select the function of the PE3/TIOC0D/SCK |
| 12 | PE3MD0 | 0 | R/W | 000: PE3 I/O (port) |
| | | | | 001: TIOC0D I/O (MTU2) |
| | | | | 110: SCK0 I/O (SCI) |
| | | | | Other than above: Setting prohibited |
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value always be 0. |
| 10 | PE2MD2 | 0 | R/W | PE2 Mode |
| 9 | PE2MD1 | 0 | R/W | Select the function of the PE2/TIOC0C/TXD |
| 8 | PE2MD0 | 0 | R/W | 000: PE2 I/O (port) |
| | | | | 001: TIOC0C I/O (MTU2) |
| | | | | 110: TXD0 output (SCI) |
| | | | | Other than above: Setting prohibited |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value always be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | 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 |

| Bit | Bit Name | Value | R/W | Description |
|---|---|---|---|---|
| 15 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write v<br>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 1<br>(PE15MD2 to PE15MD0) in PECRL4 are se<br>00: Interrupt request accept signal output<br>01: Setting prohibited<br>10: Interrupt request accept signal output<br>11: Always high-level output |

functions. Table 20.14 shows the transmit forms of input functions allocated to se
pins. When using one of the functions shown below in multiple pins, use it with ca
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, $\overline{\text{WAIT}}$, $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, PO<br>$\overline{\text{POE5}}$ |

OR type: Signals input to several pins are formed as one signal through OR log
signal is transmitted into the LSI.

AND type: Signals input to several pins are formed as one signal through AND lo
the signal is transmitted into the LSI.

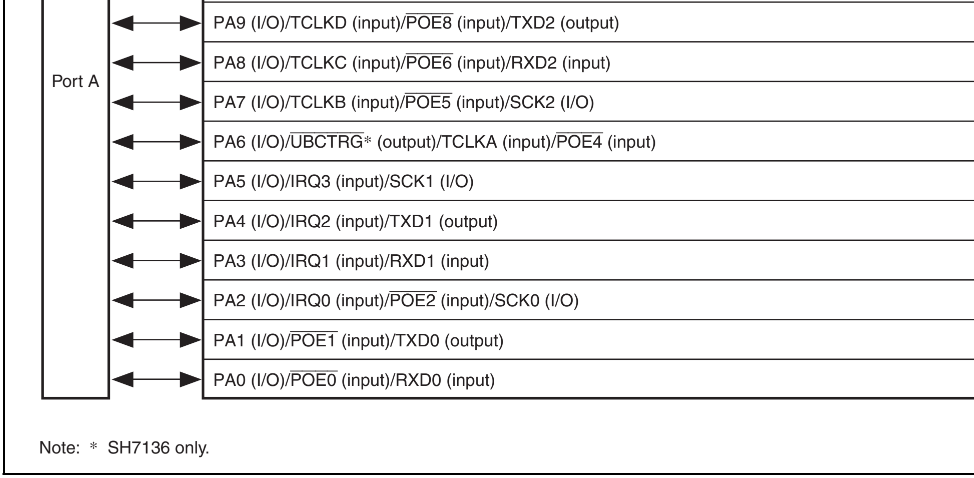
— When the pin function is output

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, co
operation cannot be guaranteed.
4. PFC setting in single-chip mode (MCU operating mode 3)

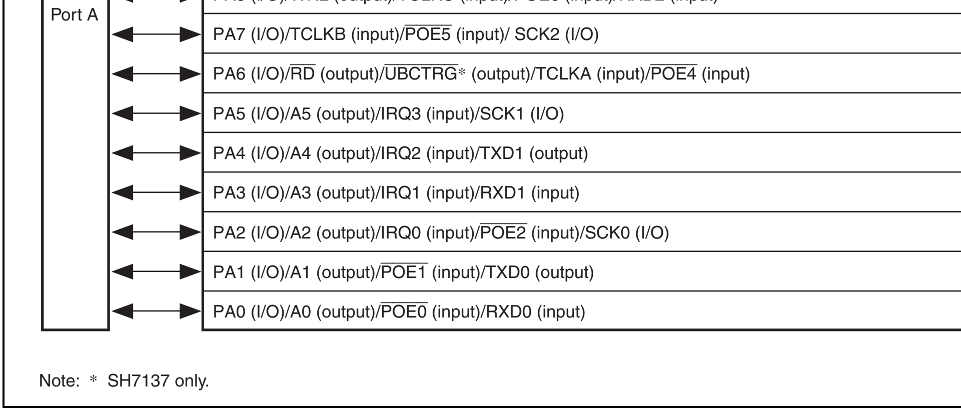
   In single-chip mode, do not set the PFC to select address bus, data bus, bus control, o
$\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, or CK signals. If they are selected, address bus signals function as hig
low-level outputs, data bus signals function as high-impedance outputs, and the other
signals function as high-level outputs. As $\overline{\text{BREQ}}$ and $\overline{\text{WAIT}}$ function as inputs, do no
them open. However, the bus-mastership-request inputs and external waits are disable

port is provided with a data register for storing the pin data.


Port A
PA9 (I/O)/TCLKD (input)/$\overline{\text{POE8}}$ (input)/TXD2 (output)
PA8 (I/O)/TCLKC (input)/$\overline{\text{POE6}}$ (input)/RXD2 (input)
PA7 (I/O)/TCLKB (input)/$\overline{\text{POE5}}$ (input)/SCK2 (I/O)
PA6 (I/O)/$\overline{\text{UBCTRG}}$* (output)/TCLKA (input)/$\overline{\text{POE4}}$ (input)
PA5 (I/O)/IRQ3 (input)/SCK1 (I/O)
PA4 (I/O)/IRQ2 (input)/TXD1 (output)
PA3 (I/O)/IRQ1 (input)/RXD1 (input)
PA2 (I/O)/IRQ0 (input)/$\overline{\text{POE2}}$ (input)/SCK0 (I/O)
PA1 (I/O)/$\overline{\text{POE1}}$ (input)/TXD0 (output)
PA0 (I/O)/$\overline{\text{POE0}}$ (input)/RXD0 (input)

Note: * SH7136 only.

**Figure 21.1 Port A (SH7131/SH7136)**


Port A
PA7 (I/O)/TCLKB (input)/$\overline{\text{POE5}}$ (input)/ SCK2 (I/O)
PA6 (I/O)/$\overline{\text{RD}}$ (output)/$\overline{\text{UBCTRG}}$* (output)/TCLKA (input)/$\overline{\text{POE4}}$ (input)
PA5 (I/O)/A5 (output)/IRQ3 (input)/SCK1 (I/O)
PA4 (I/O)/A4 (output)/IRQ2 (input)/TXD1 (output)
PA3 (I/O)/A3 (output)/IRQ1 (input)/RXD1 (input)
PA2 (I/O)/A2 (output)/IRQ0 (input)/$\overline{\text{POE2}}$ (input)/SCK0 (I/O)
PA1 (I/O)/A1 (output)/$\overline{\text{POE1}}$ (input)/TXD0 (output)
PA0 (I/O)/A0 (output)/$\overline{\text{POE0}}$ (input)/RXD0 (input)

Note: * SH7137 only.

**Figure 21.2 Port A (SH7132/SH7137)**

### 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
Bits PA15DR to PA0DR correspond to pins PA15 to PA0 (multiplexed functions omitted

When a pin function is general output, if a value is written to PADRL, that value is outpu
from the pin, and if PADRL is read, the register value is returned directly regardless of th
state.

When a pin function is general input, if PADRL is read, the pin state, not the register valu
returned directly. If a value is written to PADRL, although that value is written into PAD
does not affect the pin state. Table 21.2 summarizes port A data register read/write operat

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PA15<br>DR | PA14<br>DR | PA13<br>DR | PA12<br>DR | PA11<br>DR | PA10<br>DR | PA9<br>DR | PA8<br>DR | PA7<br>DR | PA6<br>DR | PA5<br>DR | PA4<br>DR | PA3<br>DR | PA2<br>DR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 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 effec state |
| | Other than general input | Pin state | Can write to PADRL, but it has no effec 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 effec state |

| Bit | Bit Name | Value | R/W | Description |
|---|---|---|---|---|
| 15 | PA15PR | Pin state | R | The pin state is returned regardless of the PFC<br>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 | |

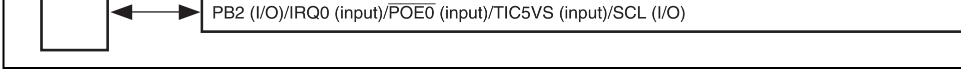

PB2 (I/O)/IRQ0 (input)/$\overline{POE0}$ (input)/TIC5VS (input)/SCL (I/O)

**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 fi

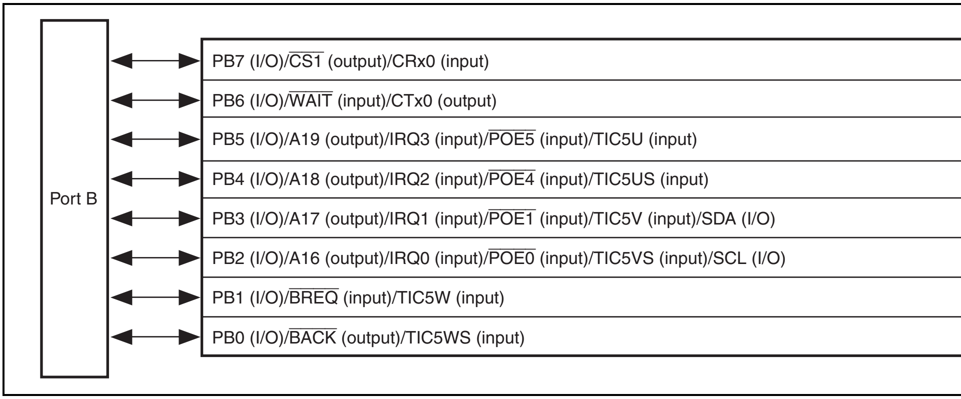

Port B
PB7 (I/O)/$\overline{CS1}$ (output)/CRx0 (input)
PB6 (I/O)/$\overline{WAIT}$ (input)/CTx0 (output)
PB5 (I/O)/A19 (output)/IRQ3 (input)/$\overline{POE5}$ (input)/TIC5U (input)
PB4 (I/O)/A18 (output)/IRQ2 (input)/$\overline{POE4}$ (input)/TIC5US (input)
PB3 (I/O)/A17 (output)/IRQ1 (input)/$\overline{POE1}$ (input)/TIC5V (input)/SDA (I/O)
PB2 (I/O)/A16 (output)/IRQ0 (input)/$\overline{POE0}$ (input)/TIC5VS (input)/SCL (I/O)
PB1 (I/O)/$\overline{BREQ}$ (input)/TIC5W (input)
PB0 (I/O)/$\overline{BACK}$ (output)/TIC5WS (input)

**Figure 21.4 Port B (SH7132/SH7137)**

### 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 Bits PB7DR to PB2DR correspond to pins PB7 to PB2, respectively (multiplexed functio omitted here) in the SH7131 and SH7136. Bits PB7DR to PB0DR correspond to pins PB 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 outpu from the pin, and if PBDRL is read, the register value is returned directly regardless of th state.

When a pin function is general input, if PBDRL is read, the pin state, not the register valu returned directly. If a value is written to PBDRL, although that value is written into PBD does not affect the pin state. Table 21.4 summarizes port B data register read/write operat

| | | | | |
|---|---|---|---|---|
| 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 va 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 o state |
| | Other than general input | Pin state | Can write to PBDRL, but it has no effect o 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 o state |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

| 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 va always be 0. |
| 7 | PB7PR | Pin state | R | The pin state is returned regardless of the PF 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 va always be 0. |

- PBPRL (SH7132/SH7137)

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | PB7PR | PB6PR | PB5PR | PB4PR | PB3PR | PB2PR |
| 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 |

| | | | |
|---|---|---|---|
| 2 | PB2PR | Pin state | R |
| 1 | PB1PR | Pin state | R |
| 0 | PB0PR | Pin state | R |

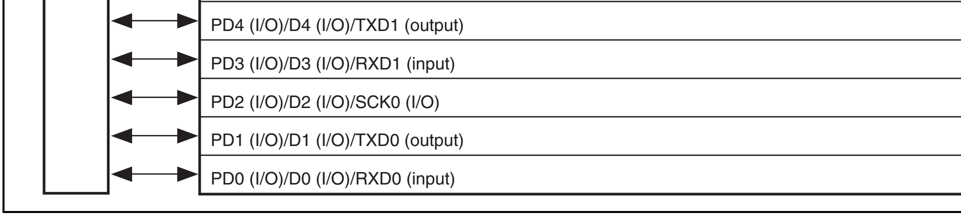

PD4 (I/O)/D4 (I/O)/TXD1 (output)
PD3 (I/O)/D3 (I/O)/RXD1 (input)
PD2 (I/O)/D2 (I/O)/SCK0 (I/O)
PD1 (I/O)/D1 (I/O)/TXD0 (output)
PD0 (I/O)/D0 (I/O)/RXD0 (input)

**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 anc Port D has the following registers. For details on register addresses and register states du processing, refer to section 25, List of Registers.

**Table 21.5 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Ac |
|---|---|---|---|---|---|
| Port D data register L | PDDRL | R/W | H'0000 | H'FFFFD282 | 8, |
| Port D port register L | PDPRL | R | H'xxxx | H'FFFFD29E | 8, |

does not affect the pin state. Table 21.6 summarizes port D data register read/write operat

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | PB7 DR | PB7 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 |
| R/W: | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

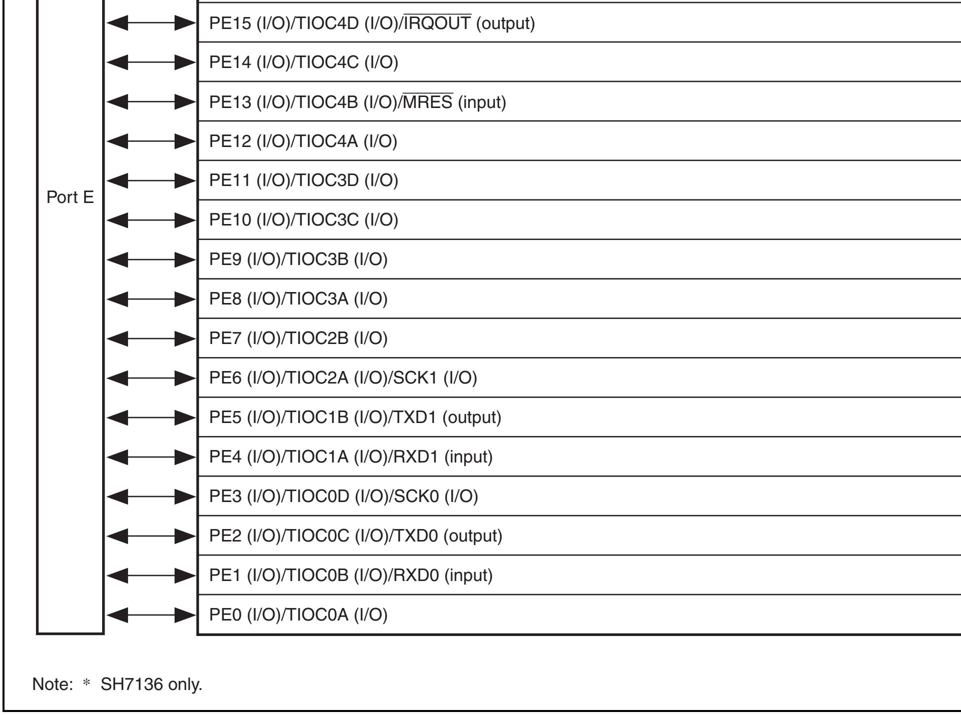
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu always be 0. |
| 10 | PD10DR | 0 | R/W | See table 21.6. |
| 9 | PD9DR | 0 | R/W | |
| 8 | PD8DR | 0 | R/W | |
| 7 | PD7DR | 0 | R/W | |
| 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 | |

### 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 the pins regardless of the PFC setting. Bits PD10PR to PD0PR correspond to pins PD10 (multiplexed functions omitted here).

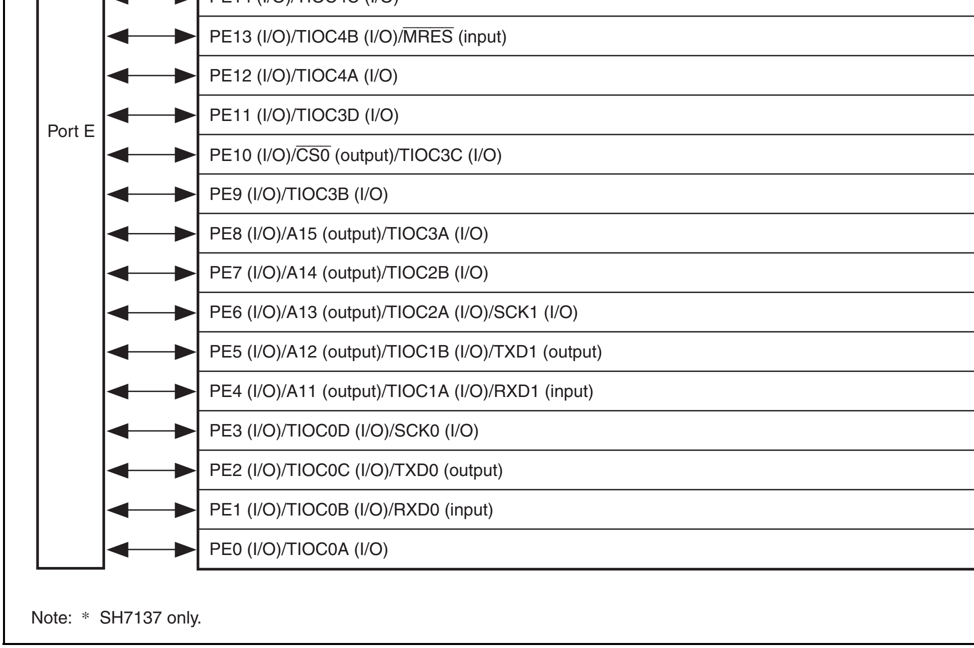
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | PD10<br>PR | PD9<br>PR | PD8<br>PR | PD7<br>PR | PD6<br>PR | PD5<br>PR | PD4<br>PR | PD3<br>PR | PD2<br>PR |
| Initial value: | 0 | 0 | 0 | 0 | 0 | * | * | * | * | * | * | * | * | * |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | 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 |


Port E
PE15 (I/O)/TIOC4D (I/O)/$\overline{\text{IRQOUT}}$ (output)
PE14 (I/O)/TIOC4C (I/O)
PE13 (I/O)/TIOC4B (I/O)/$\overline{\text{MRES}}$ (input)
PE12 (I/O)/TIOC4A (I/O)
PE11 (I/O)/TIOC3D (I/O)
PE10 (I/O)/TIOC3C (I/O)
PE9 (I/O)/TIOC3B (I/O)
PE8 (I/O)/TIOC3A (I/O)
PE7 (I/O)/TIOC2B (I/O)
PE6 (I/O)/TIOC2A (I/O)/SCK1 (I/O)
PE5 (I/O)/TIOC1B (I/O)/TXD1 (output)
PE4 (I/O)/TIOC1A (I/O)/RXD1 (input)
PE3 (I/O)/TIOC0D (I/O)/SCK0 (I/O)
PE2 (I/O)/TIOC0C (I/O)/TXD0 (output)
PE1 (I/O)/TIOC0B (I/O)/RXD0 (input)
PE0 (I/O)/TIOC0A (I/O)

Note: * SH7136 only.

**Figure 21.6 Port E (SH7131/SH7136)**


PE14 (I/O)/TIOC4C (I/O)
PE13 (I/O)/TIOC4B (I/O)/MRES (input)
PE12 (I/O)/TIOC4A (I/O)
Port E
PE11 (I/O)/TIOC3D (I/O)
PE10 (I/O)/CS0 (output)/TIOC3C (I/O)
PE9 (I/O)/TIOC3B (I/O)
PE8 (I/O)/A15 (output)/TIOC3A (I/O)
PE7 (I/O)/A14 (output)/TIOC2B (I/O)
PE6 (I/O)/A13 (output)/TIOC2A (I/O)/SCK1 (I/O)
PE5 (I/O)/A12 (output)/TIOC1B (I/O)/TXD1 (output)
PE4 (I/O)/A11 (output)/TIOC1A (I/O)/RXD1 (input)
PE3 (I/O)/TIOC0D (I/O)/SCK0 (I/O)
PE2 (I/O)/TIOC0C (I/O)/TXD0 (output)
PE1 (I/O)/TIOC0B (I/O)/RXD0 (input)
PE0 (I/O)/TIOC0A (I/O)

Note: * SH7137 only.

**Figure 21.7 Port E (SH7132/SH7137)**

| | | | | | |
|---|---|---|---|---|---|
| Port E port register H | PEPRH | R | H'00xx | H'FFFFD31C | 8, |
| Port E port register L | PEPRL | R | H'xxxx | H'FFFFD31E | 8, |

### 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 reg 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 v output directly from the pin, and if PEDRH or PEDRL is read, the register value is retur directly regardless of the pin state.

When a pin function is general input, if PEDRH or PEDRL is read, the pin state, not the value, is returned directly. If a value is written to PEDRH or PEDRL, although that valu written into PEDRH or PEDRL, it does not affect the pin state. Table 21.8 summarizes p register read/write operations.

| | | | | |
|---|---|---|---|---|
| 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 | |

| | | | |
|---|---|---|---|
| 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 |

general output PEDRL value 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 tha return the states of the pins regardless of the PFC setting. Bits PE21PR to PE0PR corresp 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | PE21 PR | PE20 PR | PE19 PR | PE18 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 |

| 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 valu always be 0. |
| 5 | PE21PR | Pin state | R | The pin state is returned regardless of the PFC 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 | |

|  |  |  |  |
|---|---|---|---|
| 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 |

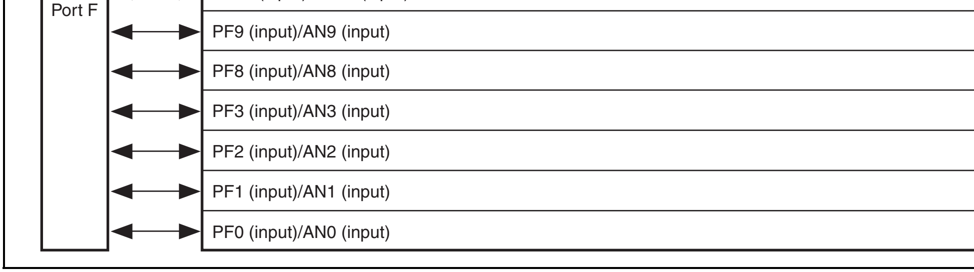


**Figure 21.8 Port F (SH7131/SH7136)**

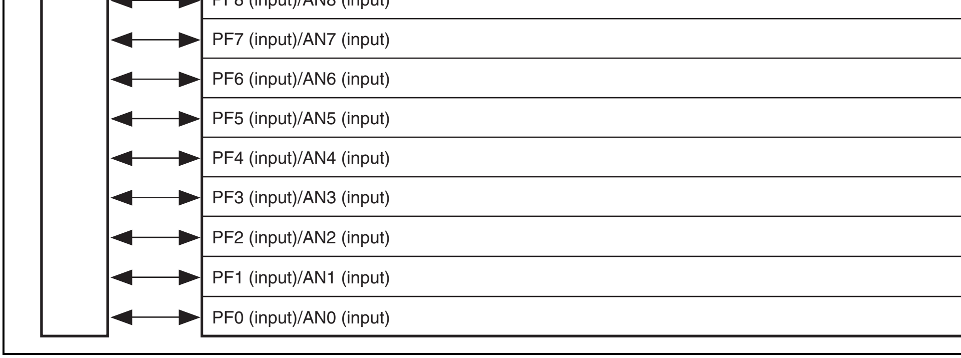


**Figure 21.9 Port F (SH7132/SH7137)**

### 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 SH7132 and SH7137. Port F has the following register. For details on register addresses 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 | Ac |
|---|---|---|---|---|---|
| Port F data register L | PFDRL | R | H'xxxx | H'FFFFD382 | 8, |

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PF15<br>DR | PF14<br>DR | PF13<br>DR | PF12<br>DR | PF11<br>DR | PF10<br>DR | PF9<br>DR | PF8<br>DR | - | - | - | - | PF3<br>DR | PF2<br>DR |
| Initial value: | * | * | * | * | * | * | * | * | 0 | 0 | 0 | 0 | * | * |
| R/W: | 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 valu 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 | |

| | | | |
|---|---|---|---|
| 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) |

memory MAT that is currently mapped. The MAT can be switched by bank-switchin
LSI has started up.

- — Size of the user MAT, from which booting-up proceeds after a power-on reset in mode: 256 Kbytes or 128 Kbytes
- — Size of the user boot MAT, from which booting-up proceeds after a power-on res 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 MAT or user-boot MAT can be programmed, and the bit rate for data transfer betwe and this LSI are automatically adjusted.

  **User Program Mode**: This mode allows programming of the user MAT via any des interface.

  **User Boot Mode**: This mode allows writing of a user boot program via any desired i and programming of the user MAT.

  **Off-board programming mode**

  **Programmer Mode:** This mode allows programming of the user MAT and user boo with the aid of a PROM programmer.

- Downloading of an on-chip program to provide an interface for programming/erasur

  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 setti parameters as arguments. “User branching” is also supported.

There are two modes of protection: software protection is applied by register settings hardware protection is applied by the level on the FWE pin. Protection of the flash me from programming or erasure can be selected.

When an abnormal state is detected, such as runaway execution of programming/erasi protection modes initiate the transition to the error protection state and suspend programming/erasing processing.

- Programming/erasing time

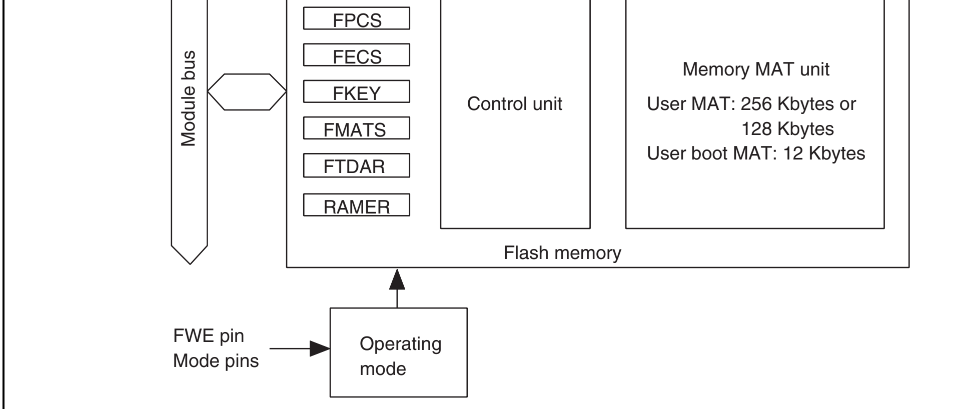
  The time taken to program 128 bytes of flash memory in a single round is $t_P$ ms (typ.) equivalent to $t_P$/128 ms per byte. The erasing time is $t_E$s (typ.) per block.

- 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).



[Legend]

| | |
|---|---|
| FCCS: | Flash code control and status register |
| FPCS: | Flash program code select register |
| FECS: | Flash erase code select register |
| FKEY: | Flash key code register |
| FMATS: | Flash MAT select register |
| FTDAR: | Flash transfer destination address register |
| RAMER: | RAM emulation register |

**Figure 22.1 Block Diagram of Flash Memory**

user boot mode, and boot mode.

- Flash memory can be read, programmed, or erased by means of the PROM programm
programmer mode.

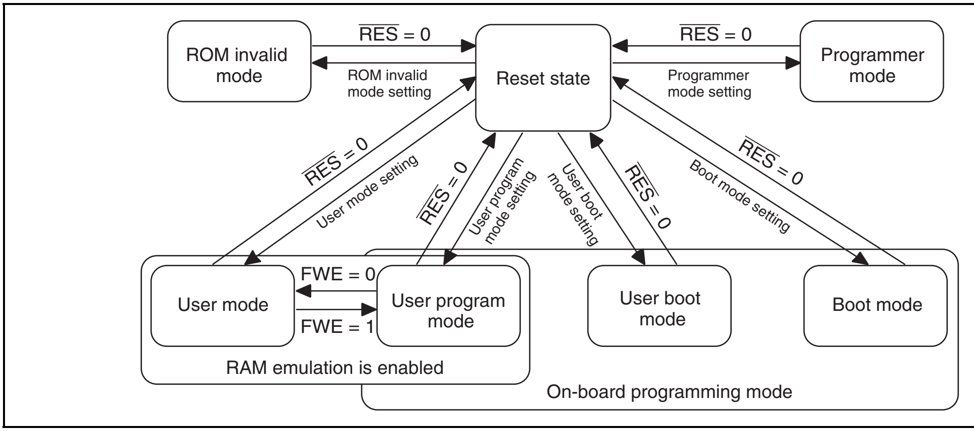


**Figure 22.2 Mode Transition of Flash Memory**

Note: External bus extended mode and user boot mode are not supported by the SH71
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 | Progr Mode |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RES}}$ | 0 | 1 | 1 | 1 | 1 | 1 | Settin depen condit specia PROM progra |
| 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
be used)

| | | | | |
|---|---|---|---|---|
| Programming/ erasing enable MAT | User MAT<br>User boot MAT | User MAT | User MAT | User MA<br>User boo |
| Programming/ erasing control | Command method | Programming/ erasing interface | Programming/ erasing interface | — |
| All erasure | Possible (Automatic) | Possible | Possible | Possible (Automa |
| Block division erasure | Possible*[1] | Possible | Possible | Not poss |
| Program data transfer | From host via SCI | From optional device via RAM | From optional device via RAM | Via prog |
| User branch function | Not possible | Possible | Possible | Not poss |
| RAM emulation | Not possible | Possible | Not possible | Not poss |
| Reset initiation MAT | Embedded program storage MAT | User MAT | User boot MAT*[2] | Embedde program 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 f 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 programme
- The user MAT and user boot MAT are all erased in boot mode. Then, the user MAT a boot MAT can be programmed by means of the command method. However, the cont the MAT cannot be read until this state.

  Only user boot MAT is programmed and the user MAT is programmed in user boot m 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 setting different from user program mode.

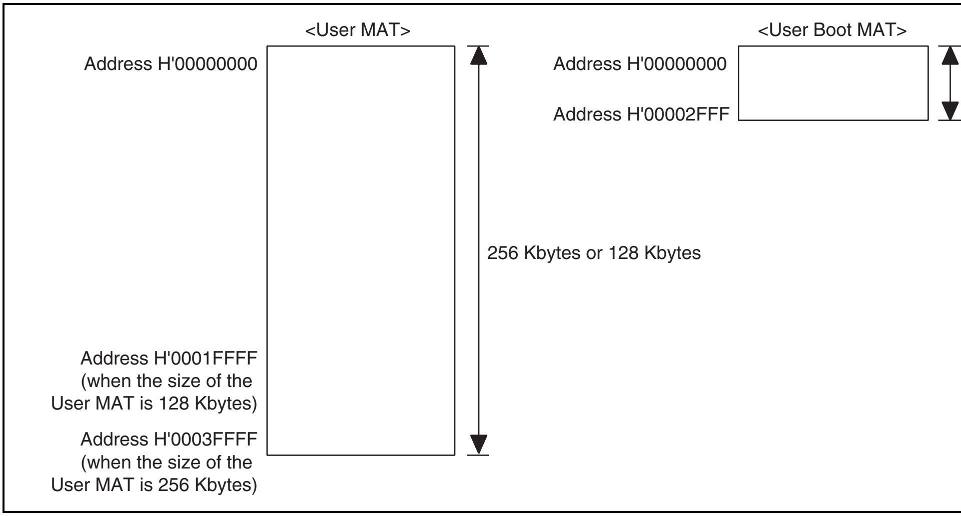
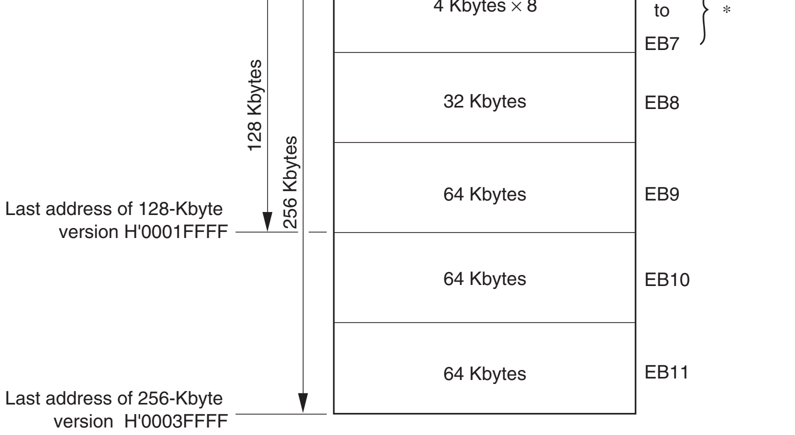

<User MAT>
Address H'00000000
Address H'0001FFFF (when the size of the User MAT is 128 Kbytes)
Address H'0003FFFF (when the size of the User MAT is 256 Kbytes)
256 Kbytes or 128 Kbytes
<User Boot MAT>
Address H'00000000
Address H'00002FFF

**Figure 22.3 Flash Memory Configuration**

The user MAT and user boot MAT have different memory sizes. Do not access a user b
that is 12 Kbytes or more. When a user boot MAT exceeding 12 Kbytes is read from, an
value is read.


4 Kbytes × 8
to
EB7
*
128 Kbytes
256 Kbytes
32 Kbytes
EB8
64 Kbytes
EB9
Last address of 128-Kbyte version H'0001FFFF
64 Kbytes
EB10
64 Kbytes
EB11
Last address of 256-Kbyte version H'0003FFFF

Note: * RAM emulation can be performed in the eight 4-Kbyte blocks.

**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 RA specifying the program address/data and erase block by using the interface registers/para

The procedure program is made by the user in user program mode and user boot mode. T overview of the procedure is as follows. For details, see section 22.5.2, User Program Mo

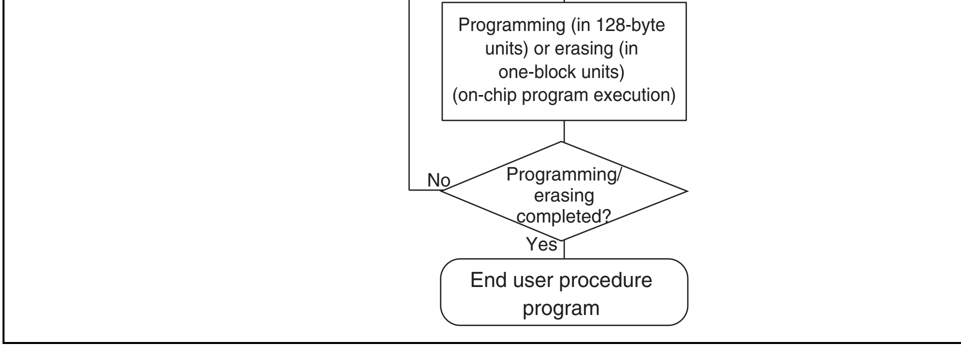

Programming (in 128-byte units) or erasing (in one-block units) (on-chip program execution)
No
Programming/ erasing completed?
Yes
End user procedure program

**Figure 22.5 Overview of User Procedure Program**

(1) Selection of On-Chip Program to be Downloaded and Setting of Download Destinat

This LSI has programming/erasing programs and they can be downloaded to the on-c RAM. The on-chip program to be downloaded is selected by setting the correspondin the programming/erasing interface registers. The download destination can be specif FTDAR.

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/era
The user branch destination must be in an area other than the user MAT area which is
middle of programming and the area where the on-chip program is downloaded. Thes
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
entered.

The program data/programming destination address is specified in 128-byte units whe
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 ar
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
execution result is returned to the programming/erasing interface parameters.

The area to be programmed must be erased in advance when programming flash mem

There are limitations and notes on the interrupt processing during programming/erasir
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,
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 processin
download and initialization are not required when the same processing is executed
consecutively.

| | | | |
|---|---|---|---|
| Mode 0* | MD0 | Input | Sets operating mode of this L |
| Transmit data | TXD1 (PA4) | Output | Serial transmit data output (u<br>boot mode) |
| Receive data | RXD1 (PA3) | Input | Serial receive data input (use<br>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 shown in table 22.4.

There are several operating modes for accessing flash memory, for example, read mode/ mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/p are allocated for each operating mode and MAT selection. The correspondence of opera and registers/parameters for use is shown in table 22.5.

| | | | | |
|---|---|---|---|---|
| Flash transfer destination address register | FTDAR | R/W | H'00 | H'FFFFCC06 |
| RAM emulation register | RAMER | R/W | H'0000 | H'FFFFF108 |

Notes: 1. The bits except the SCO bit are read-only bits. The SCO bit is a programming-
(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**

| | | | Initial | | Ac |
|---|---|---|---|---|---|
| **Name** | **Abbreviation** | **R/W** | **Value** | **Address** | **Si** |
| Download pass/fail result | DPFR | R/W | Undefined | On-chip RAM* | 8, |
| Flash pass/fail result | FPFR | R/W | Undefined | R0 of CPU | 8, |
| Flash multipurpose address area | FMPAR | R/W | Undefined | R5 of CPU | 8, |
| Flash multipurpose data destination area | FMPDR | R/W | Undefined | R4 of CPU | 8, |
| Flash erase block select | FEBS | R/W | Undefined | R4 of CPU | 8, |
| Flash program and erase frequency control | FPEFEQ | R/W | Undefined | R4 of CPU | 8, |
| Flash user branch address set parameter | FUBRA | R/W | Undefined | R5 of CPU | 8, |

Note: * One byte of the start address in the on-chip RAM area specified by FTDAR is v

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 r

2. The setting may be required according to the combination of initiation mode a target MAT.

| | 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 t performs hardware protection of the flash memor 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 prot 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 MA 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 shou be 0. |

| | | | | |
|---|---|---|---|---|
| | | | | 0: Flash memory operates normally<br>Programming/erasing protection for flash mer<br>protection) is invalid.<br>[Clearing condition]<br>At a power-on reset<br>1: Indicates an error occurs during programming<br>flash memory.<br>Programming/erasing protection for flash mer<br>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<br>always be 0. |

Four NOP instructions must be executed immedia
setting this bit to 1.

For interrupts during download, see section 22.8.
Interrupts during Programming/Erasing. For the d
time, see section 22.8.3, Other Notes.

Since this bit is cleared to 0 when download is co
this bit cannot be read as 1.

Download by setting the SCO bit to 1 requires a s
interrupt processing that performs bank switching
on-chip program storage area. Therefore, before
download request (SCO = 1), set VBR to H'8400
Otherwise, the CPU gets out of control. Once dow
end is confirmed, VBR can be changed to any oth

The mode in which the FWE pin is high must be u
when using the SCO function.

0: Download of the on-chip programming/erasing
to the on-chip RAM is not executed.

[Clearing condition]

When download is completed

1: Request that the on-chip programming/erasing
is downloaded to the on-chip RAM is generate

[Setting conditions]

When all of the following conditions are satisfied a
written to this bit

- FKEY is written to H'A5
- During execution in the on-chip RAM
- Not in RAM emulation mode (RAMS in RAMC

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| | | | | These bits are always read as 0. The write value<br>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<br>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 |

| 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<br>When a value other than H'A5 is written to FKEY, cannot be written to the SCO bit. Therefore down to the on-chip RAM cannot be executed.<br>Only when H'5A is written, programming/erasing memory can be executed. Even if the on-chip programming/erasing program is executed, flash cannot be programmed or erased when a value o H'5A is written to FKEY.<br>H'A5: Writing to the SCO bit is enabled (The SCO cannot be set by a value other than H'A5.)<br>H'5A: Programming/erasing is enabled (A value o H'5A enables software protection state.)<br>H'00: Initial value |

| | | | | |
|---|---|---|---|---|
| 6 | MS6 | 0 | R/W | These bits are in user-MAT selection state wher other than H'AA is written and in user-boot-MAT state when H'AA is written.<br>The MAT is switched by writing a value in FMAT on-chip RAM instrunction.<br>When the MAT is switched, follow section 22.8. Switching between User MAT and User Boot M user boot MAT cannot be programmed in user p mode if user boot MAT is selected by FMATS. T boot MAT must be programmed in boot mode o programmer mode.)<br>H'AA: The user boot MAT is selected (in user-M selection state when the value of these bi other than H'AA)<br>Initial value when these bits are initiated i boot mode.<br>H'00: Initial value when these bits are initiated i except for user boot mode (in user-MAT s state)<br>[Programmable condition]<br>These bits are in the execution state in the on-c |
| 5 | MS5 | 0/1 | R/W | |
| 4 | MS4 | 0 | R/W | |
| 3 | MS3 | 0/1 | R/W | |
| 2 | MS2 | 0 | R/W | |
| 1 | MS1 | 0/1 | R/W | |
| 0 | MS0 | 0 | R/W | |

| Bit | Bit Name | 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 c start address set by bits 6 to 0 (TDA6 to TDA0). the address setting is erroneous or not is tested checking whether the setting of TDA6 to TDA0 i range of H'00 to H'04 after setting the SCO bit ir to 1 and performing download. Before setting th bit to 1 be sure to set the FTDAR value betweer 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 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 from H'00 to H'04 can be set to specify the dowr start address in on-chip RAM in 2-Kbyte units.<br>A value from H'05 to H'7F cannot be set. If such is set, the TDER bit (bit 7) in this register is set t prevent download from being executed.<br>H'00: Download start address is set to H'FFFF9(<br>H'01: Download start address is set to H'FFFF98<br>H'02: Download start address is set to H'FFFFA<br>H'03: Download start address is set to H'FFFFA<br>H'04: Download start address is set to H'FFFFB<br>H'05 to H'7F: Setting prohibited. If this value is s TDER bit (bit 7) is set to 1 to abort download processing. |

must be saved at the processing start. (The maximum size of a stack area to be used is 1

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 b have different meanings according to the processing program. See the description of FP each processing.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| erasing frequency control | | | | | | | | |
| Flash user branch address set | FUBRA | — | √ | — | — | R/W | Undefined | R5 |
| Flash multipurpose address area | FMPAR | — | — | √ | — | R/W | Undefined | R5 |
| Flash multipurpose data destination area | FMPDR | — | — | √ | — | R/W | Undefined | R4 |
| Flash erase block select | FEBS | — | — | — | √ | R/W | Undefined | R4 |

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 RAM area to be downloaded is the area as much as 3 Kbytes starting from the start ad 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 value is given by the DPFR parameter.

(a) Download pass/fail result parameter (DPFR: one byte of start address of on-chip R 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 performed by setting one byte of the start address of the on-chip RAM area specifi FTDAR to a value other than the return value of download (for example, H'FF) be download start (before setting the SCO bit to 1). For the checking method of down results, see section 22.5.2 (2), Programming Procedure in User Program Mode.

| | | | | |
|---|---|---|---|---|
| | | | | specified as only one type. When more than t of the program are selected, the program is n selected, or the program is selected without n an error occurs.<br>0: Download program can be selected norma<br>1: Download error occurs (Multi-selection or p 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 set to H'A5.<br>0: FKEY setting is normal (FKEY = H'A5)<br>1: FKEY setting is abnormal (FKEY = value o H'A5) |
| 0 | SF | Undefined | R/W | Success/Fail<br>Returns the result whether download has end normally or not.<br>0: Downloading on-chip program has ended r (no error)<br>1: Downloading on-chip program has ended a (error occurs) |

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 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 |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F15 | F14 | F13 | F12 | F11 | F10 | F9 | F8 | F7 | F6 | F5 | F4 | F3 | F2 |
| 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 |

digit and is written to the FPEFEQ paramet (general register R4). For example, when t operating frequency of the CPU is 28.882 M value is as follows.

- The number to three decimal places o rounded and the value is thus 28.88.
- The formula that 28.88 × 100 = 2888 i converted to the binary digit and B'000 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 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | UA31 | UA30 | UA29 | UA28 | UA27 | UA26 | UA25 | UA24 | UA23 | UA22 | UA21 | UA20 | UA19 | UA18 |
| 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 |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | UA15 | UA14 | UA13 | UA12 | UA11 | UA10 | UA9 | UA8 | UA7 | UA6 | UA5 | UA4 | UA3 | UA2 |
| 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 |

program download area and stack area must n
overwritten. If CPU runaway occurs or the dow
area or stack area is overwritten, the value of f
memory cannot be guaranteed.

The download of the on-chip program, initializa
initiation of the programming/erasing program 
be executed in the processing of the user bran
destination. Programming or erasing cannot be
guaranteed when returning from the user branc
destination. The program data which has alrea
prepared must not be programmed.

Store general registers R8 to R15. General reg
to R7 are available without storing them.

Moreover, the programming/erasing interface r
must not be written to or RAM emulation mode
be entered in the processing of the user branch
destination.

After the processing of the user branch has en
programming/erasing program must be returne
using the RTS instruction.

For the execution intervals of the user branch
processing, see note 2 (User branch processin
intervals) in section 22.8.3, Other Notes.

| 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 specifie branch destination address is in the area othe storage area of the programming/erasing prog 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 specifie operating frequency of the CPU is in the rang 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 n<br>0: Initialization has ended normally (no error)<br>1: Initialization has ended abnormally (error o |

data must be in the consecutive space, which can be accessed by using the MOV.I 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 da 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 general register R4. This parameter is called FMPDR (flash multipurpose data des area parameter).

For details on the programming procedure, see section 22.5.2, User Program Mod

(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

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. I boundary condition is not satisfied, an error occurs. The error occurrence is indicated WA bit (bit 1) in FPFR.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MOA31 | MOA30 | MOA29 | MOA28 | MOA27 | MOA26 | MOA25 | MOA24 | MOA23 | MOA22 | MOA21 | MOA20 | MOA19 | MOA18 | MO |
| 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 |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MOA15 | MOA14 | MOA13 | MOA12 | MOA11 | MOA10 | MOA9 | MOA8 | MOA7 | MOA6 | MOA5 | MOA4 | MOA3 | MOA2 | M |
| 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 |

This parameter indicates the start address in the area, which stores the data to be prog in the user MAT. When the storage destination of the program data is in flash memo 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MOD31 | MOD30 | MOD29 | MOD28 | MOD27 | MOD26 | MOD25 | MOD24 | MOD23 | MOD22 | MOD21 | MOD20 | MOD19 | MOD18 |
| 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 |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MOD15 | MOD14 | MOD13 | MOD12 | MOD11 | MOD10 | MOD9 | MOD8 | MOD7 | MOD6 | MOD5 | MOD4 | MOD3 | MOD2 |
| 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 |

| 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 stor program data for the user MAT. The consecu byte data is programmed to the user MAT sta the specified start address. |

| 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 Dete<br>Returns the check result of whether the signal the FWE pin is high and whether the error prot state is not entered.<br>When a low-level signal is input to the FWE pir error protection state is entered, 1 is written to The input level to the FWE pin and the error pr state can be confirmed with the FWE bit (bit 7) FLER bit (bit 4) in FCCS, respectively. For con enter the error protection state, see section 22 Error Protection.<br>0: FWE and FLER settings are normal (FWE = = 0)<br>1: FWE = 0 or FLER = 1, and programming car performed |

| | | | | |
|---|---|---|---|---|
| | | | | If FMATS is set to H'AA and the user boot MA<br>selected, an error occurs when programming<br>performed. In this case, both the user MAT ar<br>boot MAT are not rewritten.<br>Programming of the user boot MAT must be e<br>in boot mode or programmer mode.<br>0: Programming has ended normally<br>1: Programming has ended abnormally (progr<br>result is not guaranteed) |
| 4 | FK | Undefined | R/W | Flash Key Register Error Detect<br>Returns the check result of the value of FKEY<br>the start of the programming processing.<br>0: FKEY setting is normal (FKEY = H'5A)<br>1: FKEY setting is error (FKEY = value other t |
| 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<br>as the start address of the storage destination<br>program data, an error occurs.<br>0: Setting of write data address is normal<br>1: Setting of write data address is abnormal |

| | | | | |
|---|---|---|---|---|
| | | | | 1: Setting of programming destination address abnormal |
| 0 | SF | Undefined | R/W | Success/Fail<br>Indicates whether the program processing has normally or not.<br>0: Programming has ended normally (no error)<br>1: Programming has ended abnormally (error c |

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 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 |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | 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 |

| 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 f 11. 0 corresponds to the EB0 block and 1 corresponds to the EB11 block. An error o when a number other than 0 to 11 (H'00 t set.<br>• 128-Kbyte flash memory<br>Set the erase-block number in the range f 0 corresponds to the EB0 block and 9 cor to the EB9 block. An error occurs when a other than 0 to 9 (H'00 to H'09) is set. |

| 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 the FWE pin is high and whether the error prot state is not entered.<br>When a low-level signal is input to the FWE pir error protection state is entered, 1 is written to The input level to the FWE pin and the error pr state can be confirmed with the FWE bit (bit 7) FLER bit (bit 4) in FCCS, respectively. For con enter the error protection state, see section 22 Error Protection.<br>0: FWE and FLER settings are normal (FWE = = 0)<br>1: FWE = 0 or FLER = 1, and erasure cannot b performed |

| | | | | |
|---|---|---|---|---|
| | | | | selected, an error occurs when erasure is per<br>this case, both the user MAT and user boot M<br>not erased.<br>Erasure of the user boot MAT must be execut<br>mode or programmer mode.<br>0: Erasure has ended normally<br>1: Erasure has ended abnormally (erasure res<br>guaranteed) |
| 4 | FK | Undefined | R/W | Flash Key Register Error Detect<br>Returns the check result of FKEY value befor<br>the erasing processing.<br>0: FKEY setting is normal (FKEY = H'5A)<br>1: FKEY setting is error (FKEY = value other t |
| 3 | EB | Undefined | R/W | Erase Block Select Error Detect<br>Returns the check result whether the specifie<br>block number is in the block range of the user<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<br>normally or not.<br>0: Erasure has ended normally (no error)<br>1: Erasure has ended abnormally (error occur |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R/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 valu<br>always be 0. |
| 3 | RAMS | 0 | R/W | RAM Select<br>Sets whether the user MAT is emulated or not.<br>RAMS = 1, all blocks of the user MAT are in th<br>programming/erasing protection state.<br>0: Emulation is not selected<br>Programming/erasing protection of all user-M<br>blocks is invalid<br>1: Emulation is selected<br>Programming/erasing protection of all user-M<br>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 use<br>area to be overlapped with the on-chip RAM. (S<br>22.7.) |

| | | | | |
|---|---|---|---|---|
| H'00006000 to H'00006FFF | EB6 (4 Kbytes) | 1 | 1 | 1 |
| H'00007000 to H'00007FFF | EB7 (4 Kbytes) | 1 | 1 | 1 |

Note: x: Don't care.

Boot mode executes programming/erasing user MAT and user boot MAT by means of th command and program data transmitted from the host using the on-chip SCI. The tool fo transmitting the control command and program data must be prepared in the host. The SC communication mode is set to asynchronous mode. When reset start is executed after this is set in boot mode, the boot program in the microcomputer is initiated. After the SCI bit automatically adjusted, the communication with the host is executed by means of the con command method.

The system configuration diagram in boot mode is shown in figure 22.6. For details on th setting in boot mode, see table 22.1. Interrupts are ignored in boot mode so do not genera

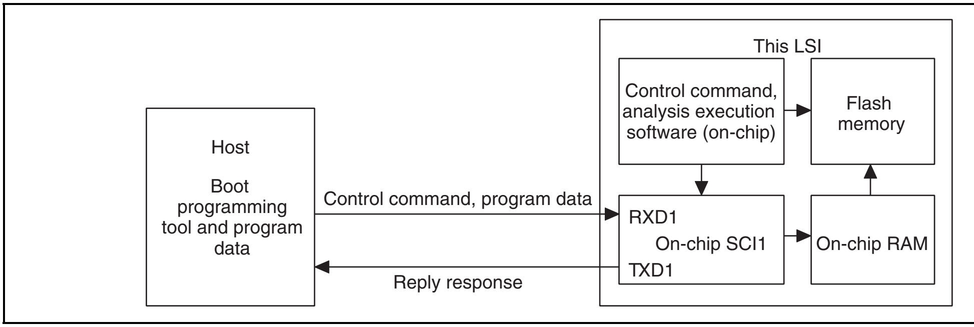

This LSI
Control command, analysis execution software (on-chip)
Flash memory
Host
Boot programming tool and program data
Control command, program data
RXD1
On-chip SCI1
TXD1
On-chip RAM
Reply response

**Figure 22.6 System Configuration in Boot Mode**

bps.

The system clock frequency which can automatically adjust the transfer bit rate of th
the bit rate of this LSI is shown in table 22.8. Boot mode must be initiated in the rang
system clock. Note that the internal clock division ratio of ×1/3 is not supported in b

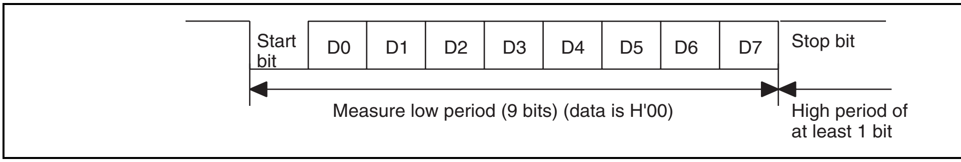

Start bit
D0
D1
D2
D3
D4
D5
D6
D7
Stop bit
Measure low period (9 bits) (data is H'00)
High period of at least 1 bit

**Figure 22.7 Automatic Adjustment Operation of SCI Bit Rate**

**Table 22.8 Peripheral Clock (Pϕ) Frequency that Can Automatically Adjust Bit R This LSI**

| Host Bit Rate | Peripheral Clock (Pϕ) Frequency Which Can Automatically Adjus 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.

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 to the programming/erasure state is sent by the host, the entire user MAT and user 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 cod followed by the address where programming should start and the data to be progra This is repeated as required while the chip is in the programming-selected state. T terminate programming, H'FFFFFFFF should be transmitted as the first address of for programming. This makes the chip return to the programming/erasure comma waiting state from the programming data waiting state.

— On receiving the erasure select command, the chip waits for the block number of a be erased. To erase a block, the host transmits the erasure command code followe number of the block to be erased. This is repeated as required while the chip is in erasure-selected state. To terminate erasure, H'FF should be transmitted as the blo number. This makes the chip return to the programming/erasure command waitin from the erasure block number waiting state. Erasure should only be executed whe specific block is to be reprogrammed without executing a reset-start of the chip af flash memory has been programmed in boot mode. If all desired programming is c single operation, such erasure processing is not necessary because all blocks are e before the chip enters the programming/erasure/other command waiting state.

— In addition to the programming and erasure commands, commands for sum check blank checking (checking for erasure) of the user MAT and user boot MAT, readi from the user MAT/user boot MAT, and acquiring current state information are p

Note that the command for reading from the user MAT/user boot MAT can only read has been programmed after automatic erasure of the entire user MAT and user boot M

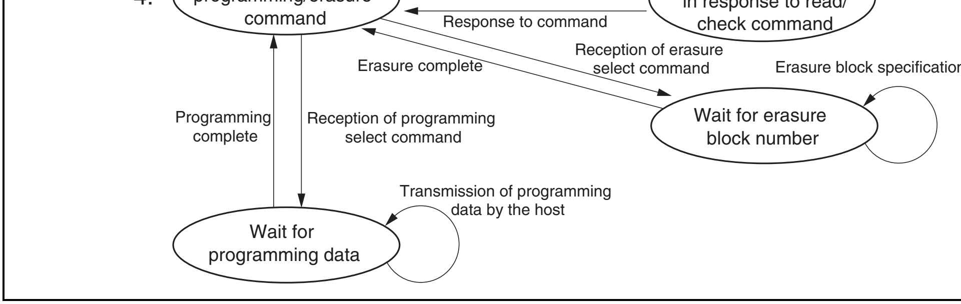
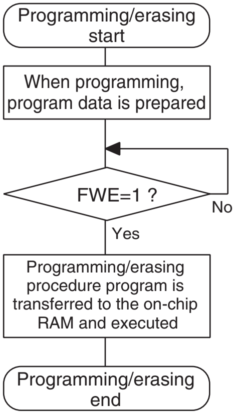


**Figure 22.8 State Transitions in Boot Mode**

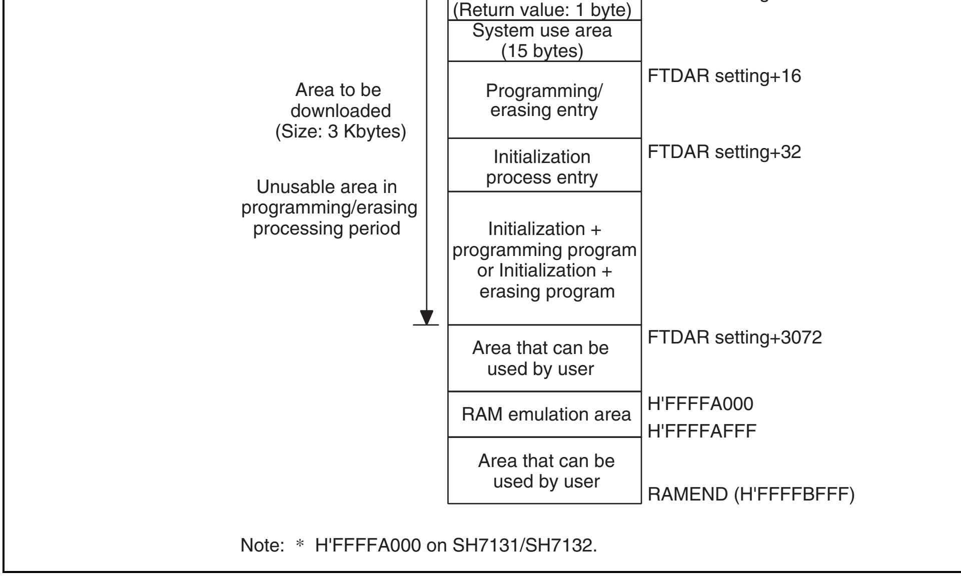
period, which is longer than the normal 100 µs.

For details on the programming procedure, see the description in section 22.5.2 (2), Progr Procedure in User Program Mode. For details on the erasing procedure, see the descriptio section 22.5.2 (3), Erasing Procedure in User Program Mode.


Programming/erasing start
When programming, program data is prepared
FWE=1 ?
No
Yes
Programming/erasing procedure program is transferred to the on-chip RAM and executed
Programming/erasing end

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**


(Return value: 1 byte)
System use area (15 bytes)
FTDAR setting+16
Area to be downloaded (Size: 3 Kbytes)
Programming/ erasing entry
FTDAR setting+32
Initialization process entry
Unusable area in programming/erasing processing period
Initialization + programming program or Initialization + erasing program
FTDAR setting+3072
Area that can be used by user
H'FFFFA000
RAM emulation area
H'FFFFAFFF
Area that can be used by user
RAMEND (H'FFFFBFFF)

Note: * H'FFFFA000 on SH7131/SH7132.

**Figure 22.10 RAM Map after Download**

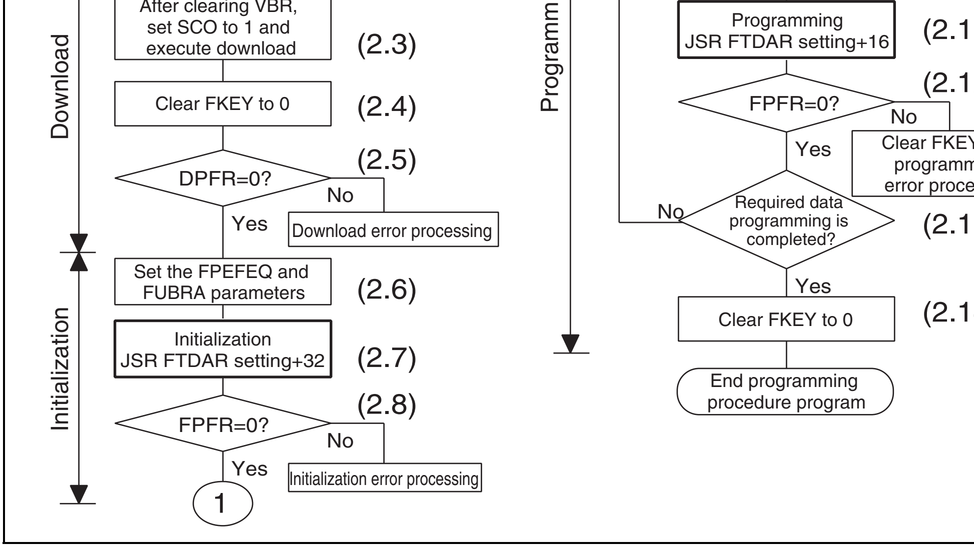

Download
After clearing VBR, set SCO to 1 and execute download
(2.3)
Clear FKEY to 0
(2.4)
(2.5)
DPFR=0?
No
Yes
Download error processing
Initialization
Set the FPEFEQ and FUBRA parameters
(2.6)
Initialization JSR FTDAR setting+32
(2.7)
(2.8)
FPFR=0?
No
Yes
Initialization error processing
1
Programm
Programming JSR FTDAR setting+16
(2.1
(2.1
FPFR=0?
No
Yes
Clear FKEY
programm
error proce
No
Required data programming is completed?
(2.1
Yes
Clear FKEY to 0
(2.1
End programming procedure program

**Figure 22.11 Programming Procedure**

The details of the programming procedure are described below. The procedure progra be executed in an area other than the flash memory to be programmed. Especially the where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-ch Specify 1/4 (initial value) as the frequency division ratios of an internal clock (Iϕ), a b (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 clea the setting of the frequency control register (FRQCR) can be changed to the desired v

The area that can be executed in the steps of the user procedure program (on-chip RA MAT, and external space) is shown in section 22.9.2, Areas for Storage of the Proced Program and Data for Programming.

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 pro set, download is not performed and a download error is returned to the source select 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 download request.

(2.3) VBR is set to 0 and 1 is written to the SCO bit of FCCS, and then download is exe

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 ret user procedure program, the SCO bit is cleared to 0. Therefore, the SCO bit cannot b confirmed to be 1 in the user procedure program.

The download result can be confirmed only by the return value of the DPFR paramet the SCO bit is set to 1, incorrect decision must be prevented by setting the DPFR par that is one byte of the start address of the on-chip RAM area specified by FTDAR, t other than the return value (H'FF).

When download is executed, particular interrupt processing, which is accompanied b switch as described below, is performed as an internal microcomputer processing, sc need to be set to H'84000000. Four NOP instructions are executed immediately after instructions that set the SCO bit to 1.

- The user MAT space is switched to the on-chip program storage area.

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 b before setting the SCO bit to 1.

If flash memory is accessed by the DTC during downloading, operation cannot be gu 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 down destination specified by FTDAR). If the value is H'00, download has been perf normally. If the value is not H'00, the source that caused download to fail can investigated by the description below.
- If the value of the DPFR parameter is the same as before downloading (e.g. H' address setting of the download destination in FTDAR may be abnormal. In th confirm the setting of the TDER bit (bit 7) in FTDAR.
- If the value of the DPFR parameter is different from before downloading, chec bit (bit 2) and the FK bit (bit 1) in the DPFR parameter to ensure that the dow program selection and FKEY register setting were normal, respectively.

(2.6) The operating frequency is set to the FPEFEQ parameter and the user branch destin set to the FUBRA parameter for initialization.

other than the one that is to be programmed. The area of the on-chip program

downloaded cannot be set.

The program processing must be returned from the user branch processing by

instruction.

See the description in section 22.4.3 (2.2), Flash user branch address setting p

(FUBRA: general register R5 of CPU).

(2.7) Initialization

When a programming program is downloaded, the initialization program is also dow
on-chip RAM. There is an entry point of the initialization program in the area from (
start address set by FTDAR) + 32 bytes. The subroutine is called and initialization is
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 maxi
  bytes must be reserved in RAM.
- Interrupts can be accepted during the execution of the initialization program.
  the program storage area and stack area in on-chip RAM and register values
  be destroyed.

(2.8) The return value of the initialization program, FPFR (general register R0) is check

(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.

If the storage destination of the program data is flash memory, even when the p execution routine is executed, programming is not executed and an error is retu the FPFR parameter. In this case, the program data must be transferred to on-c and then programming must be executed.

(2.11) Programming

There is an entry point of the programming program in the area from (download start set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and programmin 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 bytes must be reserved in RAM.

(2.12) The return value in the programming program, FPFR (general register R0) is check

(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 byte units, and repeat steps (2.10) to (2.13). Increment the programming destination a 128 bytes and update the programming data pointer correctly. If an address which has been programmed is written to again, not only will a programming error occur, but als 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 finished, secure a reset period (period of $\overline{\text{RES}}$ = 0) that is at least as long as the norma

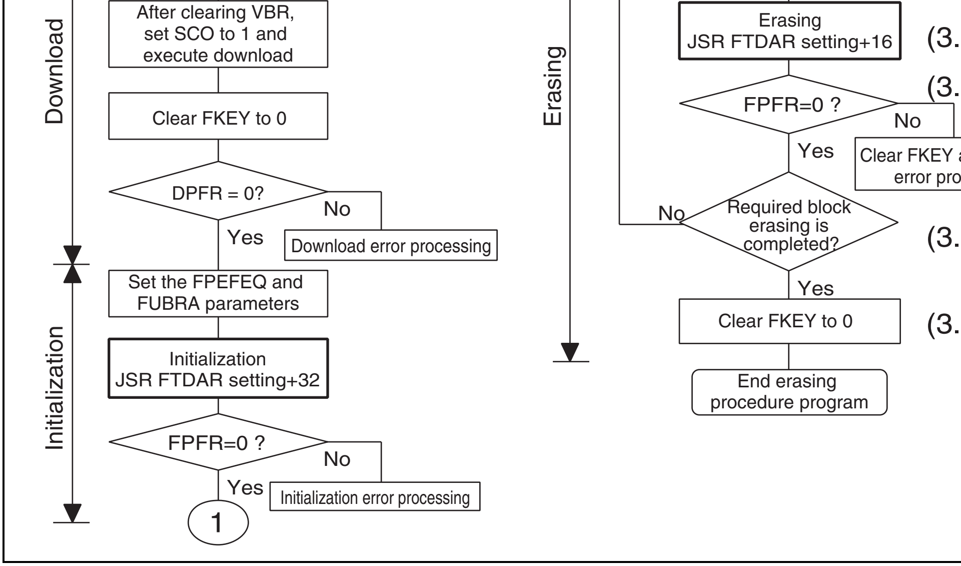

Download
After clearing VBR, set SCO to 1 and execute download
Clear FKEY to 0
DPFR = 0?
No
Yes
Download error processing
Set the FPEFEQ and FUBRA parameters
Initialization
Initialization JSR FTDAR setting+32
FPFR=0 ?
No
Yes
Initialization error processing
1
Erasing
Erasing JSR FTDAR setting+16
(3.
FPFR=0 ?
(3.
No
Yes
Clear FKEY a error pro
No
Required block erasing is completed?
(3.
Yes
Clear FKEY to 0
(3.
End erasing procedure program

**Figure 22.12 Erasing Procedure**

The details of the erasing procedure are described below. The procedure program mu executed in an area other than the user MAT to be erased. Especially the part where bit in FCCS is set to 1 for downloading must be executed in on-chip RAM. Specify value) as the frequency division ratios of an internal clock (If), a bus clock (Bf), and peripheral clock (Pf) through the frequency control register (FRQCR).

After the programming/erasing program has been downloaded and the SCO bit is cle the setting of the frequency control register (FRQCR) can be changed to the desired

The area that can be executed in the steps of the user procedure program (on-chip RA MAT, and external space) is shown in section 22.9.2, Areas for Storage of the Proce Program and Data for Programming.

Specify the start address of the download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, same as those in the programming procedure. For details, see the description in sectio (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 general register R4). If a value other than an erase block number of the user MAT is s block is erased even though the erasing program is executed, and an error is returned return value parameter FPFR.

(3.3) Erasure

Similar to as in programming, there is an entry point of the erasing program in the are (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine i and erasing is executed by using the following steps.

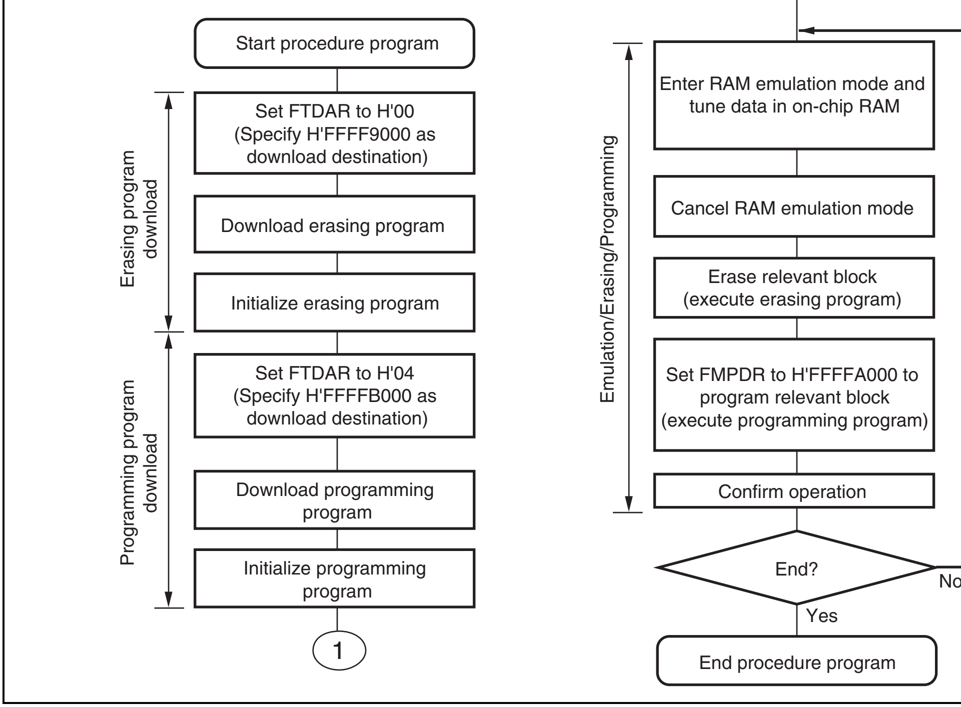
```
MOV.L  #DLTOP+16,R1         ; Set entry address to R1
JSR    @R1                  ; Call erasing routine
NOP
```

- 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 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 (3.5). Blocks that have already been erased can be erased again.



Start procedure program
Erasing program download
Set FTDAR to H'00 (Specify H'FFFF9000 as download destination)
Download erasing program
Initialize erasing program
Programming program download
Set FTDAR to H'04 (Specify H'FFFFB000 as download destination)
Download programming program
Initialize programming program
1
Emulation/Erasing/Programming
Enter RAM emulation mode and tune data in on-chip RAM
Cancel RAM emulation mode
Erase relevant block (execute erasing program)
Set FMPDR to H'FFFFA000 to program relevant block (execute programming program)
Confirm operation
End?
No
Yes
End procedure program

**Figure 22.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Prog (Overview)**

In the above example, the erasing program and programming program are download 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:

### 22.5.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings than thos
program mode or boot mode. User boot mode is a user-arbitrary boot mode, unlike boot r
uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing
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-memor
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
and all other interrupts cannot be accepted. This period is 100 μs while operating at ar
frequency of 40 MHz.

Next, processing starts from the execution start address of the reset vector in the user
MAT. At this point, H'AA is set to the flash MAT select register (FMATS) because th
execution MAT is the user boot MAT.

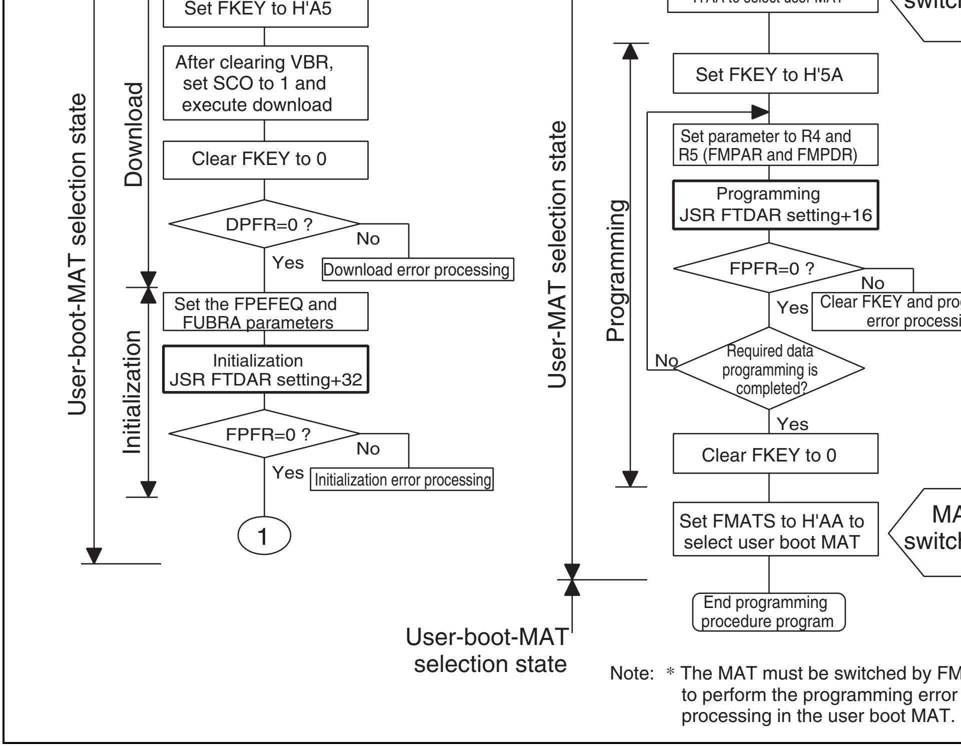

Set FKEY to H'A5
After clearing VBR, set SCO to 1 and execute download
Clear FKEY to 0
DPFR=0 ?
No
Yes
Download error processing
Set the FPEFEQ and FUBRA parameters
Initialization JSR FTDAR setting+32
FPFR=0 ?
No
Yes
Initialization error processing
1
User-boot-MAT selection state
Download
Initialization
User-MAT selection state
Programming
H'AA to select user MAT
switc
Set FKEY to H'5A
Set parameter to R4 and R5 (FMPAR and FMPDR)
Programming JSR FTDAR setting+16
FPFR=0 ?
No
Yes
Clear FKEY and prog error processi
No
Required data programming is completed?
Yes
Clear FKEY to 0
Set FMATS to H'AA to select user boot MAT
MA switcl
End programming procedure program
User-boot-MAT selection state
Note: * The MAT must be switched by FM to perform the programming error processing in the user boot MAT.

**Figure 22.14 Procedure for Programming User MAT in User Boot Mod**

MAT the interrupt vector is read from is undetermined. Perform MAT switching in ac with the description in section 22.8.1, Switching between User MAT and User Boot M

Except for MAT switching, the programming procedure is the same as that in user pro mode.

The area that can be executed in the steps of the user procedure program (on-chip RA MAT, and external space) is shown in section 22.9.2, Areas for Storage of the Proced Program and Data for Programming.

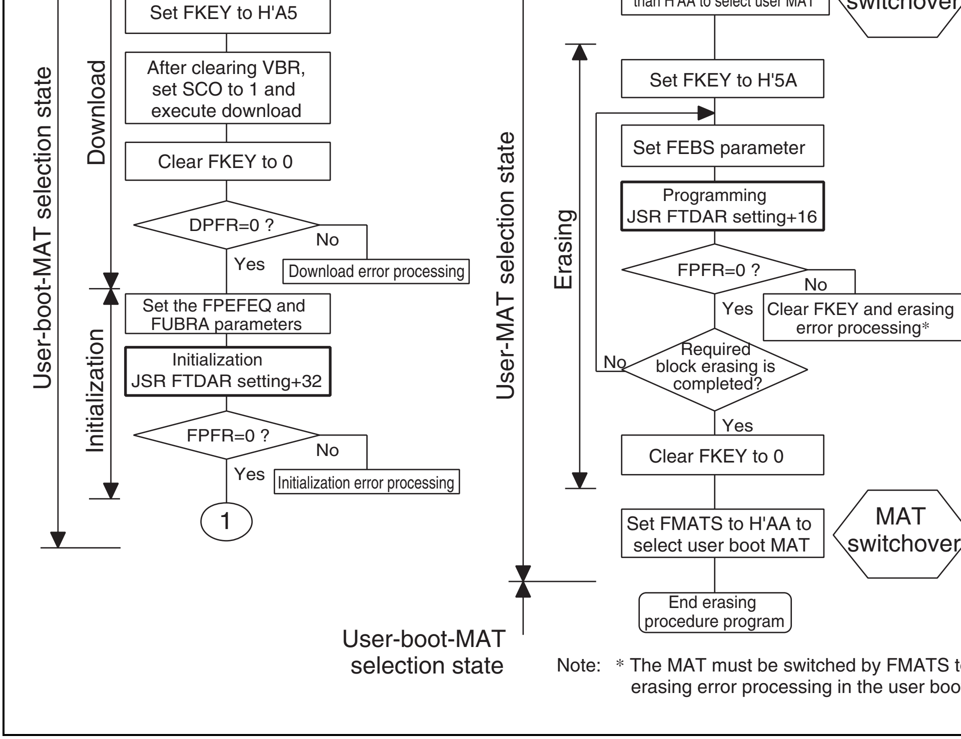

Set FKEY to H'A5
After clearing VBR, set SCO to 1 and execute download
Clear FKEY to 0
DPFR=0 ?
No
Yes
Download error processing
Set the FPEFEQ and FUBRA parameters
Initialization JSR FTDAR setting+32
FPFR=0 ?
No
Yes
Initialization error processing
1
Download
Initialization
User-boot-MAT selection state
than H'AA to select user MAT
switchover
Set FKEY to H'5A
Set FEBS parameter
Programming JSR FTDAR setting+16
FPFR=0 ?
No
Yes
Clear FKEY and erasing error processing*
No
Required block erasing is completed?
Yes
Clear FKEY to 0
Set FMATS to H'AA to select user boot MAT
MAT switchover
End erasing procedure program
Erasing
User-MAT selection state
User-boot-MAT selection state
Note: * The MAT must be switched by FMATS t
erasing error processing in the user boo

**Figure 22.15 Procedure for Erasing User MAT in User Boot Mode**

The area that can be executed in the steps of the user procedure program (on-chip RAM, MAT, and external space) is shown in section 22.9.2, Areas for Storage of the Procedural and Data for Programming.

parameter.

**Table 22.9 Hardware Protection**

| | | Function to be Pr | |
|---|---|---|---|
| **Item** | **Description** | **Download** | **Prog Eras** |
| 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. | √ | √ |

| | | | |
|---|---|---|---|
| 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. | √ | √ |

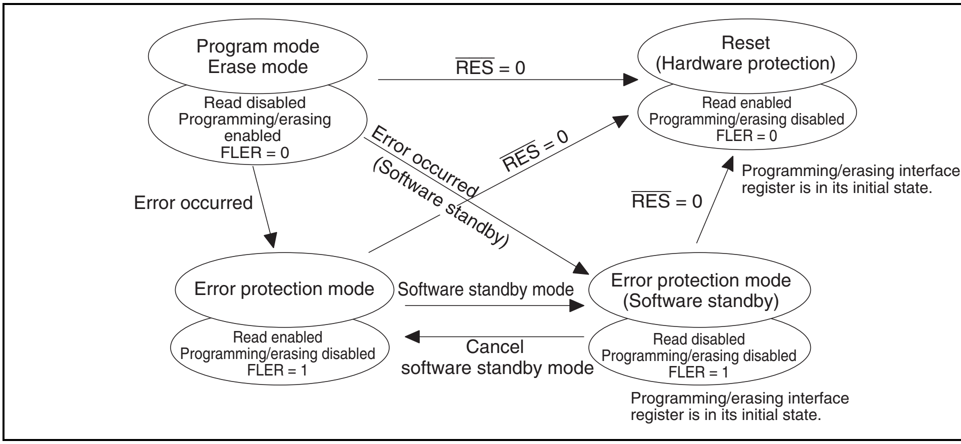
### 22.6.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occu form of the microcomputer getting out of control during programming/erasing of the flasl memory or operations that are not in accordance with the established procedures for programming/erasing. Aborting programming or erasure in such cases prevents damage t flash memory due to excessive programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, the bit in FCCS is set to 1 and the LSI enters the error protection state, thus aborting progran erasure.

memory, some voltage may still remain even after the error protection state has been en
this reason, it is necessary to reduce the risk of damage to the flash memory by extendin
period so that the charge is released.

The state-transition diagram in figure 22.16 shows transitions to and from the error prote
state.



Program mode
Erase mode
Read disabled
Programming/erasing enabled
FLER = 0
$\overline{\text{RES}}$ = 0
Reset
(Hardware protection)
Read enabled
Programming/erasing disabled
FLER = 0
Error occurred
(Software standby)
$\overline{\text{RES}}$ = 0
Programming/erasing interface register is in its initial state.
Error occurred
$\overline{\text{RES}}$ = 0
Error protection mode
Read enabled
Programming/erasing disabled
FLER = 1
Software standby mode
Cancel
software standby mode
Error protection mode
(Software standby)
Read disabled
Programming/erasing disabled
FLER = 1
Programming/erasing interface register is in its initial state.

**Figure 22.16 Transitions to and from Error Protection State**

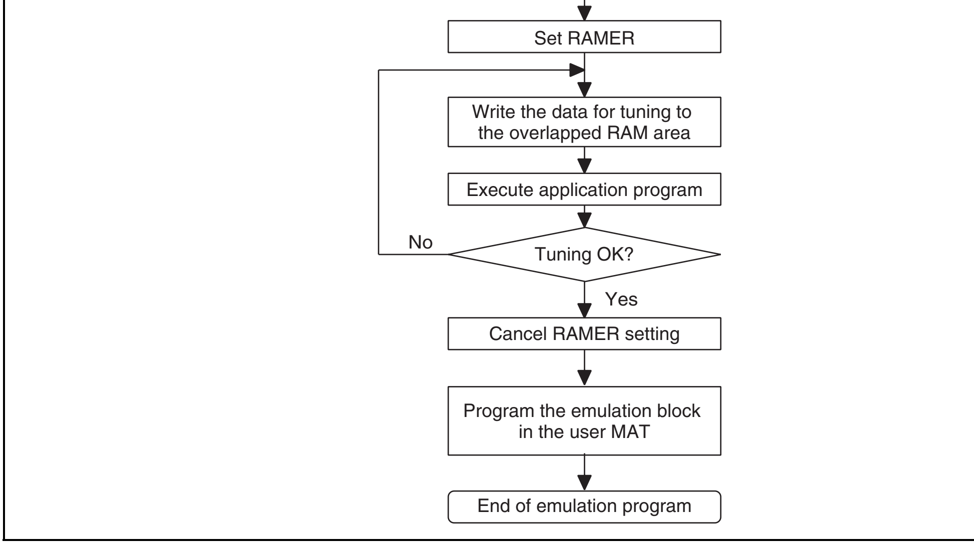

Set RAMER
Write the data for tuning to the overlapped RAM area
Execute application program
Tuning OK?
No
Yes
Cancel RAMER setting
Program the emulation block in the user MAT
End of emulation program

**Figure 22.17 Emulation of Flash Memory in RAM**

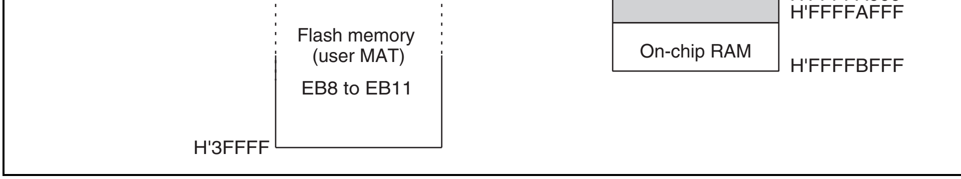


**Figure 22.18 Example of Overlapped RAM Operation (256-Kbyte Flash Memory**

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 the user MAT. The area is selected by the setting of the RAM2 to RAM0 bits in RAME

1. To overlap a part of the RAM on area EB0, to allow realtime programming of the da 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 impleme of procedural steps, including the downloading of an on-chip program. In this process, s download area with FTDAR so that the overlaid RAM area and the area where the on-c program is to be downloaded do not overlap.

Figure 22.19 shows an example of programming data that has been emulated to the EB0 the user MAT.

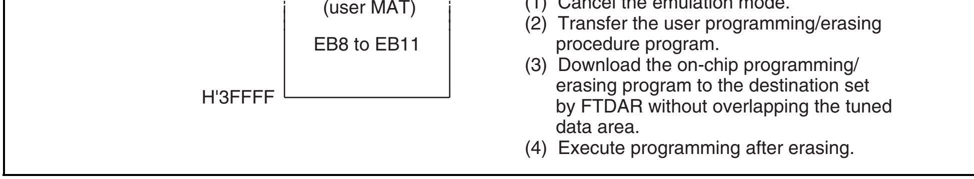

(user MAT)
EB8 to EB11
H'3FFFF
(1) Cancel the emulation mode.
(2) Transfer the user programming/erasing procedure program.
(3) Download the on-chip programming/ erasing program to the destination set by FTDAR without overlapping the tuned data area.
(4) Execute programming after erasing.

**Figure 22.19 Programming of Tuned Data (256-Kbyte Flash Memory Versio**

1. After the data to be programmed has fixed values, clear the RAMS bit to 0 to cancel t overlap of RAM. Emulation mode is canceled and emulation protection is also cleare
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 c with the download area.
4. When the EB0 area of the user MAT has not been erased, erasing must be performed programming. Set the parameters FMPAR and FMPDR so that the tuned data is desig 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 RA (emulation protection). Clear the RAMS bit to 0 before actual programming or er Though RAM emulation can also be carried out with the user boot MAT selected boot MAT can be erased or programmed only in boot mode or programmer mode

microcomputer prefetches execution instructions. Therefore, a switchover during pro
execution in the user MAT causes an instruction code in the user MAT to be prefetch
instruction in the newly selected user boot MAT to be prefetched, thus resulting in u
operation.

2. To ensure that the MAT that has been switched to is accessible, execute four NOP in
in on-chip RAM immediately after writing to FMATS of on-chip RAM (this prevent
the flash memory during MAT switching).

3. If an interrupt occurs during switching, there is no guarantee of which memory MAT
accessed.

   Always mask the maskable interrupts before switching MATs. In addition, configuri
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 wil
been switched.

   If the same interrupt processings are to be executed before and after MAT switching
interrupt requests cannot be disabled, transfer the interrupt processing routine to on-c
and use the VBR setting to place the interrupt vector table in on chip RAM. In this c
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 th
boot MAT, do not access addresses exceeding the 12-Kbyte memory space. If access
beyond the 12-Kbyte space, the values read are undefined.

(2) Write a value other than H'AA to FMATS.
(3) Execute four NOP instructions before accessing
the user MAT.

**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
value other than H'84000000, the interrupt vector table is placed in the user MAT (FM
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 tabl
or after VBR is changed is referenced may cause an error.

Therefore, for cases where VBR setting change may conflict with interrupt occurrenc
a vector table to be referenced when VBR is H'00000000 (initial value) at the start of
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
in FCCS to 1 generates a particular interrupt processing accompanied by MAT switch
Operation when the SCO download request and interrupt request conflicts is describe

1. Contention between SCO download request and interrupt request
   Figure 22.21 shows the timing of contention between execution of the instruction
   the SCO bit in FCCS to 1 and interrupt acceptance.

2. Generation of interrupt requests during downloading
   Ensure that interrupts are not generated during downloading that is initiated by th
   bit.

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 progr is to provided by the interrupt processing, temporarily save the new program data another area. After confirming the completion of programming, save the new prog in the area specified by FMPDR or change the setting in FMPDR to indicated the area in which the new program data was temporarily saved.

3. Make sure the interrupt processing routine does not rewrite the contents of the flas memory related registers or data in the downloaded on-chip program area. During interrupt processing, do not simultaneously perform RAM emulation, download o chip program by an SCO request, or programming/erasing.

4. At the beginning of the interrupt processing routine, save the CPU register content returning from the interrupt processing, write the saved contents in the CPU regist

5. When a transition is made to sleep mode or software standby mode in the interrup processing routine, the error protection state is entered and programming/erasing i aborted.

   If a transition is made to the reset state, the reset signal should only be released aft providing a reset input over a period longer than the normal 100 μs to reduce the d flash memory.

**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 t
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 SC
   FCCS that is used for a download request or FMATS that is used for MAT switchin
   sure that these registers are not accidentally written to, otherwise an on-chip progran
   downloaded and destroy RAM or a MAT switchover may occur and the CPU get ou
   control.

4. State in which interrupts are ignored

   In the following modes or period, interrupt requests are ignored; they are not execut
   interrupt sources are not retained.

   — Boot mode

   — Programmer mode

7. Monitoring runaway by WDT

   Unlike the conventional F-ZTAT SH microcomputer, no countermeasures are availab 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 interr WDT while taking the programming/erasing time into consideration as required.

1. Bit-rate matching state

   In this state, the boot program adjusts the bit rate to match that of the host. When the starts up in boot mode, the boot program is activated and enters the bit-rate matching which it receives commands from the host and adjusts the bit rate accordingly. After 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 dev mode, and bit rate are selected in this state. After making these selections, the boot p enters the programming/erasure state in response to the transition-to-programming/e state command. The boot program transfers the erasure program to RAM and execut 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 prog programming/erasure to RAM in line with the command received from the host and programming/erasure. It also performs sum checking and blank checking as directed respective commands.

Figure 22.22 shows the flow of processing by the boot program.

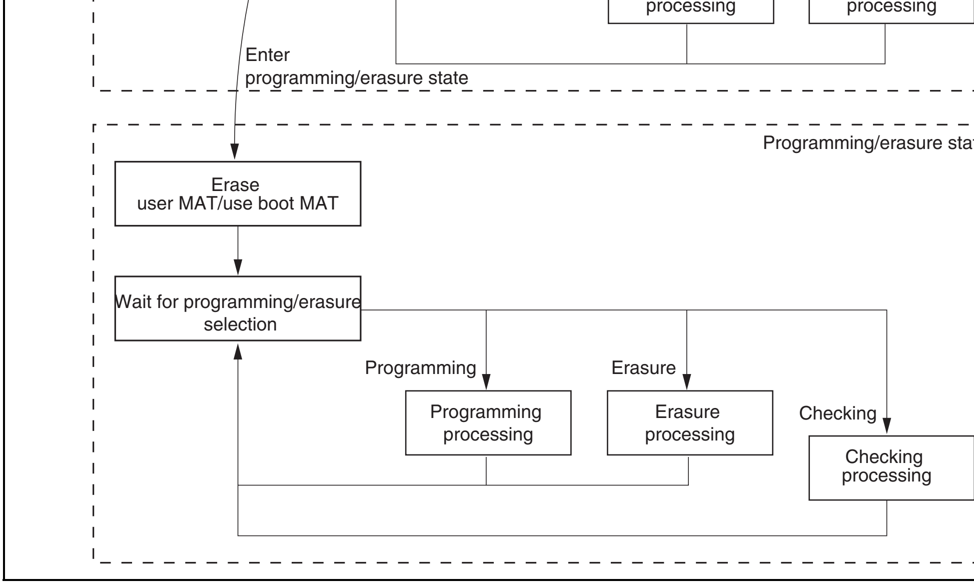


**Figure 22.22 Flow of Processing by the Boot Program**

- Bit-rate matching state

In bit-rate matching, the boot program measures the low-level intervals in a signal carryin data that is transmitted by the host, and calculates the bit rate from this. The bit rate can b changed by the new-bit-rate selection command. On completion of bit-rate matching, the program goes to the inquiry and selection state. The sequence of processing in bit-rate ma shown in figure 22.23.

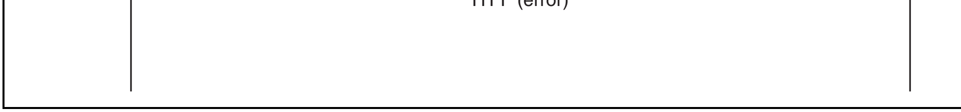


**Figure 22.23   Sequence of Bit-Rate Matching**

- Communications protocol

Formats in the communications protocol between the host and boot program after compl
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 A
   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 co
   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 respon
   error code.

4. 128-byte programming command

   The command itself does not include data-size information. The data length is know
   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.

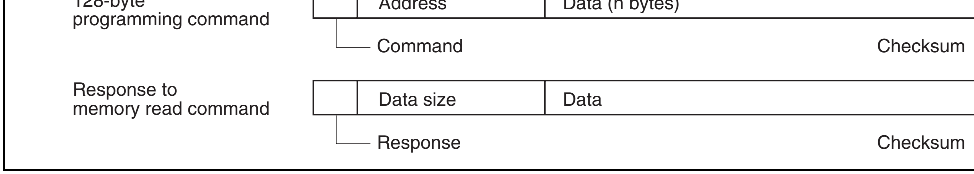


**Figure 22.24 Formats in the Communications Protocol**

— 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/re 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 co 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 co used to inquire about the programming size.

— Data size (4 bytes): Four-byte field included in the response to a memory reading command.

|  |  |  |
|---|---|---|
|  | devices | program names. |
| H'10 | Device selection | Selects a device code. |
| H'21 | Inquiry on clock modes | Requests the number of available clock modes ar 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 fr multipliers and divisors are selectable, the numbe multiplier and divisor settings for the respective cl the values of the multipliers and divisors. |
| H'23 | Inquiry on operating frequency | Requests the minimum and maximum values for frequency of the main clock and peripheral clock. |
| H'24 | Inquiry on user boot MATs | Requests the number of user boot MAT areas alo their start and end addresses. |
| H'25 | Inquiry on user MATs | Requests the number of user MAT areas along w start and end addresses. |
| H'26 | Inquiry on erasure blocks | Requests the number of erasure blocks along wit 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 era 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) mode selection (H'11), new bit rate selection (H'3F). These commands are mandatory. If selection command is sent two or more times, the command that is sent last is effective.

| | |
|---|---|
| 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 fie and the checksum. Here, it is the total number of bytes taken up by the number of number of characters, device code, and product name fields.

— Number of devices (1 byte): The number of device models supported by the boot p embedded in the microcomputer.

— Number of characters (1 byte): The number of characters in the device code and p 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 checksu H'00.

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 sup
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 avail
modes.

| | |
|---|---|
| Command | H'21 |

— Command H'21 (1 byte): Inquiry on clock modes

| | | | | | |
|---|---|---|---|---|---|
| Response | H'31 | Size | Mode | … | SUM |

|  |  |  |  |  |
|---|---|---|---|---|
| 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 avai
  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

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 mult and multiplier fields.
- Number of operating clocks (1 byte): The number of operating clocks for which can be selected
  (for example, if frequency multiplier settings can be made for the frequencies of and peripheral operating clocks, the value should be H'02).
- Number of multipliers (1 byte): The number of multipliers selectable for the ope frequency of the main or peripheral modules
- Multiplier (1 byte):
  Multiplier: Numerical value in the case of frequency multiplication (e.g. H'04 fo
  Divisor: Two's complement negative numerical value in the case of frequency di (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 m as there are operating clocks.
- SUM (1 byte): Checksum

| | |
|---|---|
| ... | |
| 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 ar minimum values of operating frequency fields.
- Number of operating clocks (1 byte): The number of operating clock frequencies r within the device.
  For example, the value two indicates main and peripheral operating clock frequenc
- Minimum value of operating frequency (2 bytes): The minimum frequency of a fr multiplied or -divided clock signal.
  The value in this field and in the maximum value field is the frequency in MHz to decimal places, multiplied by 100 (for example, if the frequency is 20.00 MHz, th multiplied by 100 is 2000, so H'07D0 is returned here).
- Maximum value of operating frequency (2 bytes): The maximum frequency of a fr multiplied or -divided clock signal.
  As many pairs of minimum/maximum values are included as there are operating c
- SUM (1 byte): Checksum

| | |
|---|---|
| … | |
| 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 fi
- 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 M 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 | | |

In response to the inquiry on erasure blocks, the boot program returns the number of erasu 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 f

— 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

— 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 rat the new bit rate and, if the setting was successful, responds to the ACK sent by the host 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 multipli 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 set H'00C0, which is 192 in decimal notation).

— Input frequency (2 bytes): The frequency of the clock signal fed to the boot progr
This should be the frequency in MHz to the second decimal place, multiplied by example, if the frequency is 28.882 MHz, the values is truncated to the second de place and multiplied by 100, making 2888; so H'0B48 should be set in this field)

Divisor: Two's complement negative numerical value in the case of frequency div
(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 f
minimum to the maximum value).

H'26: Frequency multiplier error (the specified multiplier does not match an availa

H'27: Operating frequency error (the specified operating frequency is not within th
from the minimum to the maximum value).

3. Operating frequency

The operating frequency is calculated from the received input frequency and the freq multiplier or divisor. The input frequency is the frequency of the clock signal suppli LSI, while the operating frequency is the frequency at which the LSI is actually driv 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 m and maximum values of the operating frequency for the selected clock mode of the s 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 select bits (CKS) in the serial mode register (SCSMR) and the value (= N) of the bit register (SCBRR) are calculated, after which the error in the bit rate is calculated. Th checked to see if it is smaller than 4%. A result greater than or equal to 4% generates selection error. The following formula is use to calculate the error.

$$\text{Error (\%)} = \left\{ \left[ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} \right] - 1 \right\} \times 100$$

— Response H'06 (1 byte): The ACK code transferred in response to acknowledgeme
new bit rate

The sequence of new bit rate selection is shown in figure 22.25.

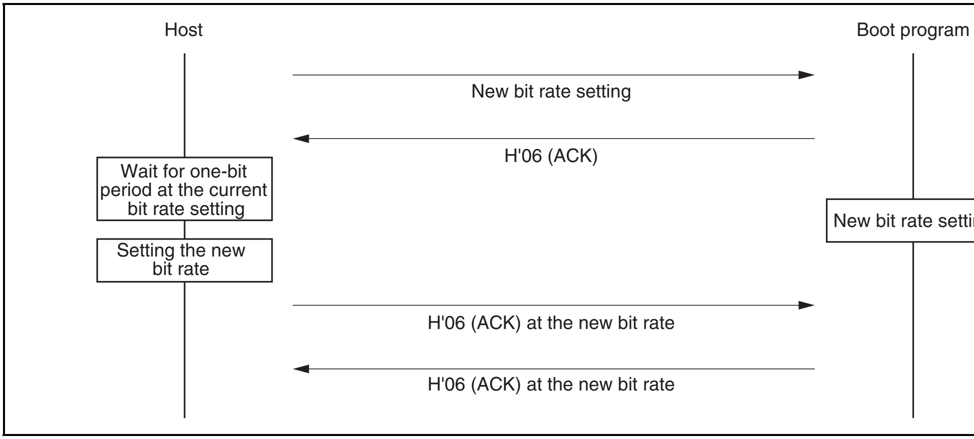

Host
Boot program
New bit rate setting
H'06 (ACK)
Wait for one-bit period at the current bit rate setting
New bit rate setti
Setting the new bit rate
H'06 (ACK) at the new bit rate
H'06 (ACK) at the new bit rate

**Figure 22.25 Sequence of New Bit Rate Selection**

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
This is returned as ACK when erasure of the user boot MAT and user MAT has s
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/er
command

— ERROR (1 byte): Error code

H'51: Erasure error (Erasure did not succeed because of an 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 de
2. Select a device from the returned device information, and send the device selection co (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 command (H'11).
5. After selection of the device and clock mode, send the commands to inquire about fre multipliers (H'22) and operating frequencies (H'23) to get the information required to new bit rate.
6. Taking into account the returned information on the frequency multipliers and operati 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 comma 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 programming/erasure state command (H'40) to place the boot program in the programming/erasure state.

| | | |
|---|---|---|
| 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 prog |
| 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 process |

Next, the host issues a 128-byte programming command. 128 bytes of data for programm the method selected by the preceding programming selection command are expected to fo command. To program more than 128 bytes, repeatedly issue 128-byte programming com To terminate programming, the host should send another 128-byte programming comman the address H'FFFFFFFF. On completion of programming, the boot program waits for the 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 co is shown in figure 22.26.

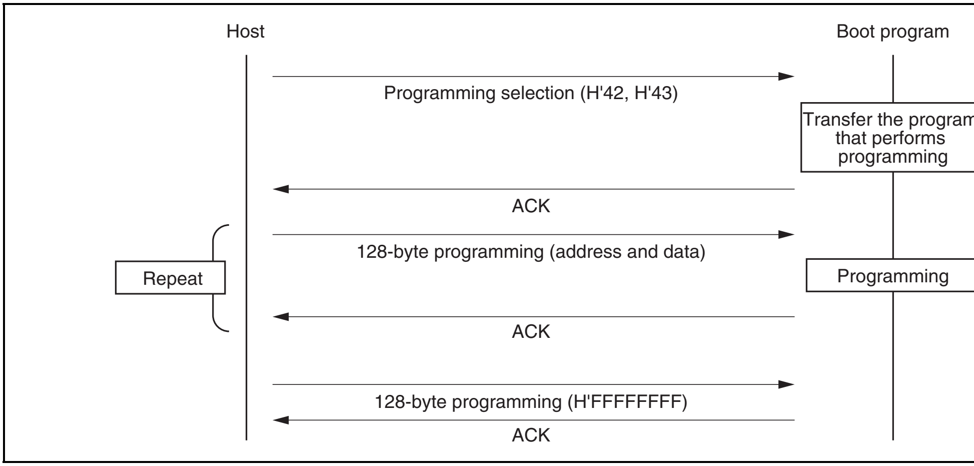

Host
Boot program
Programming selection (H'42, H'43)
Transfer the program that performs programming
ACK
128-byte programming (address and data)
Repeat
Programming
ACK
128-byte programming (H'FFFFFFFF)
ACK

**Figure 22.26 Sequence of Programming**

— 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
boot MAT.

| | | |
|---|---|---|
| Error response | H'C2 | ERROR |

— Error response H'C2 (1 byte): Error response to selection of user boot MAT prog

— ERROR (1 byte): Error code
H'54: Error in selection processing (processing was not completed because of a tr
error)

### (2) Selection of user MAT programming

In response to the command for selecting programming of the user MAT, the boot progr
transfers the corresponding flash-writing program, i.e. the program for writing to the use

| | |
|---|---|
| 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
MAT.

program transferred in response to the command to select programming of the user boot M
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 progran
  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 |

To terminate programming of a given MAT, send a 128-byte programming command w
address field H'FFFFFFFF. This informs the boot program that all data for the selected M
been sent; the boot program then waits for the next programming/erasure selection comm

| | | | |
|---|---|---|---|
| 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

in figure 22.27.

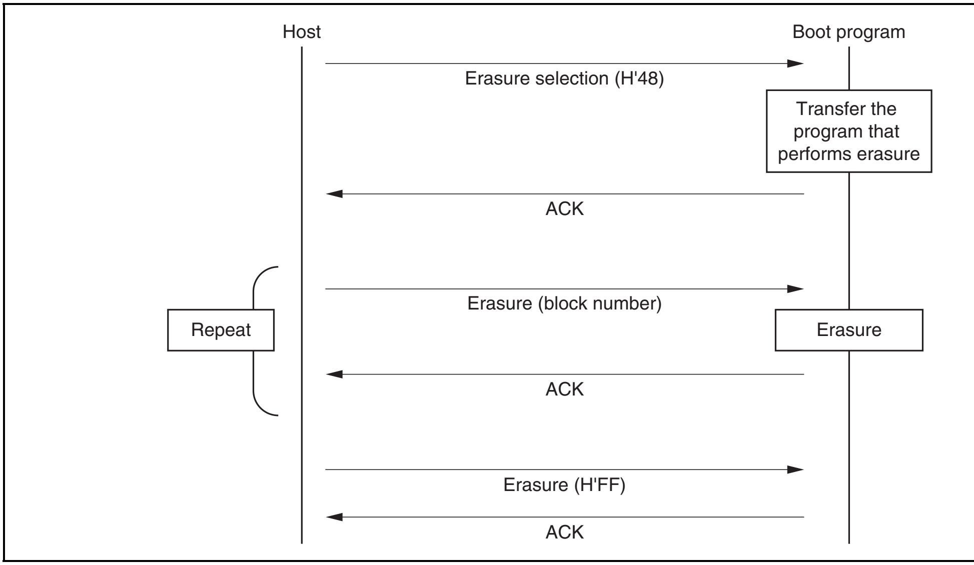


**Figure 22.27 Sequence of Erasure**

— 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 tr
error.)

### (2) Block erasure

In response to the block erasure command, the boot program erases the data in a specifie
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.

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 terminat erasure

To perform erasure again after having issued the command with the block number specifi 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 spe 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 (fixed value of 9)

| |
|---|
| 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 co
- 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
  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 o
  and the amount to read is beyond the MAT area, or "0" was specified as the amou
  read)

— 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 boo 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 al 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 MA the total of all data in the MAT, in byte units.

— SUM (1 byte): Checksum (for the transmitted data)

— 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

— 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 chec 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

— 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 operatio
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 c |
| 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) |

| | |
|---|---|
| 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 pr programs and program data are assumed to be in on-chip RAM. However, the procedure and data can be stored in and executed from other areas (e.g. external address space) as l following conditions are satisfied.

1. The on-chip programming/erasing program is downloaded from the address set by F 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. M this area is reserved.
3. Since download by setting the SCO bit to 1 will cause the MATs to be switched, it sl executed in on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, unt result of downloading has been decided. When in a mode in which the external addre

A reset state (RES = 0) for more than at least 100 µs must be taken when the LSI mod changed to reset on completion of a programming/erasing operation.

Transitions to the reset state during programming/erasing are inhibited. When the rese is accidentally input to the LSI, a longer period in the reset state than usual (100 µs) is before the reset signal is released.

7. Switching of the MATs by FMATS is needed for programming/erasing of the user M user boot mode. The program which switches the MATs should be executed from the RAM. For details, see section 22.8.1, Switching between User MAT and User Boot M 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 progra processing is within the flash memory area, an error will occur. Therefore, temporaril the program data to on-chip RAM to change the address set in FMPDR to an address than flash memory.

Based on these conditions, tables 22.17 and 22.18 show the areas in which the program d 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.

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| Programming procedure | 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

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  | (download) |  |  |  |  |
|  | Key register clearing | √ | √ | √ | √ |
|  | Judging download result | √ | √ | √ | √ |
|  | Download error processing | √ | √ | √ | √ |
|  | Setting initialization parameters | √ | √ | √ | √ |
|  | Initialization | √ | X | X | √ |
| Erasing procedure | 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 | √ | √ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Programming procedure | 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 | √ | √ | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| gramming procedure | result | | | | | |
| | 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 u

2. If the MATs have been switched by FMATS in on-chip RAM, this MAT can be u

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | register | | | | | | |
| | Writing 1 to SCO in FCCS (download) | √ | X | X | | | √ |
| | Key register clearing | √ | √ | √ | | √ | |
| | Judging download result | √ | √ | √ | | √ | |
| | Download error processing | √ | √ | √ | | √ | |
| Erasing proce- dure | 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 | √ | √ | | |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| dure | 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 u

## 22.10 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing socket adapter, just as for a discrete flash memory. Use a PROM programmer that suppor Renesas 128- or 256-Kbyte flash memory on-chip MCU device type (F-ZTATxxxx).

(CPU). Since such kind of conflict degrades the RAM access performance, software sho
created so as to avoid conflicts. For example, conflict does not occur when the buses acc
different pages. An access from the L bus (CPU) is a 1-cycle access as long as page con
not occur. The number of bus cycles in accesses from the I bus (DTC) differ depending
ratio between the internal clock (Iϕ) and bus clock (Bϕ), and the operating state of the D
contents of the on-chip RAM are retained in sleep mode or software standby mode, and
power-on reset or manual reset. However, the contents of the on-chip RAM are not retai
deep software standby mode.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the RAM co
register (RAMCR). For details on the RAM control register (RAMCR), refer to section
RAM Control Register (RAMCR).

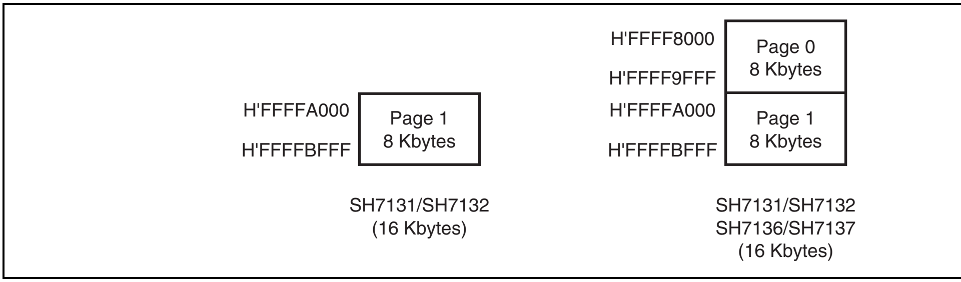


**Figure 23.1 On-chip RAM Addresses**

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 wri

### 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 the CPU and peripheral module states in each mode and the procedures for canceling ea

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| standby | instruction with STBY bit in STBCR1 and STBYMD bit in STBCR6 set to 1. | | | | (contents retained) | | IRQ<br>• Power-on the $\overline{\text{RES}}$ |
| 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 the $\overline{\text{RES}}$ |
| Module standby | Set MSTP bits in STBCR2 to STBCR5 to 1. | Runs | Runs | Held | Specified module halts (contents retained) | Specified module halts | • Clear MS<br>• Power-on modules MSTP bit initial valu |

Notes: For details on the states of on-chip peripheral module registers in each mode, refe section 25.3, Register States in Each Operating Mode. For details on the pin states mode, refer to appendix A, Pin States.

* SH7136 and SH7137 only.

| | | | | | |
|---|---|---|---|---|---|
| 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

| 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 mode<br>1: Executing SLEEP instruction makes this LSI standby mode or deep software standby mo |
| 6 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu always be 0. |

| 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 th 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 th 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 s always be 1. |
| 4 | MSTP4 | 1 | R/W | Module Stop Bit 4<br>When this bit is set to 1, the clock supply to th 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 s always be 1. |
| 2 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va always be 0. |

| | | | | |
|---|---|---|---|---|
| 7 | MSTP15 | 1 | R/W | Module Stop Bit 15<br>When this bit is set to 1, the clock supply to the halted.<br>0: I$^2$C2 operates<br>1: Clock supply to I$^2$C2 halted |
| 6 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value sh 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 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 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 is halted.<br>0: SCI_0 operates<br>1: Clock supply to SCI_0 halted |

| | | | | |
|---|---|---|---|---|
| 0 | MSTP 8 | 1 | R/W | Module Stop Bit 8<br>When this bit is set to 1, the clock supply to th<br>ET_0 is halted.<br>0: RCAN-ET_0 operates<br>1: Clock supply to RCAN-ET_0 halted |

| | | | | |
|---|---|---|---|---|
| 7 | MSTP23 | 1 | R/W | Module Stop Bit 23<br>When this bit is set to 1, the clock supply to the 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 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 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 is halted.<br>0: A/D_1 operates<br>1: Clock supply to A/D_1 halted |

### 24.3.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of modules in p
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 va always be 0. |
| 1 | MSTP25 | 1 | R/W | Module Stop Bit 25<br>When this bit is set to 1, the clock supply to th 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 th halted.<br>0: UBC operates<br>1: Clock supply to UBC halted |

| | | | | |
|---|---|---|---|---|
| 7 | AUDSRST | 0 | R/W | AUD Software Reset<br>This bit controls the AUD reset by software. W<br>written to AUDSRST, the AUD module shifts to<br>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<br>be 0. |
| 6 | HIZ | 0 | R/W | Port High-Impedance<br>In software standby mode, this bit selects whe<br>pin state is retained or changed to high-impeda<br>0: In software standby mode, the pin state is re<br>1: In software standby mode, the pin state is ch<br>to high-impedance |
| 5 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write valu<br>always be 0. |
| 1 | STBYMD | 0 | R/W | Software Standby Mode Select<br>This bit selects a transition to software standby<br>deep software standby mode by executing the<br>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 sh<br>always be 0. |

|  |  |  |  |  |
|---|---|---|---|---|
| 7 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va<br>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 th RAM is disabled. In this case, an undefined v returned when reading or fetching the data or instruction from the on-chip RAM, and writing chip RAM is ignored.<br>When RAME is cleared to 0 to disable the on- RAM, an instruction to access the on-chip RA not be set next to the instruction to write RAM such an instruction is set, normal access is no guaranteed.<br>When RAME is set to 1 to enable the on-chip instruction to read RAMCR should be set nex instruction to write to RAMCR. If an instructio access the on-chip RAM is set next to the ins write to RAMCR, normal access is not guarar |
| 3 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write va<br>always be 0. |

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 r
reset with the $\overline{\text{MRES}}$ pin, or an internal power-on/manual reset by WDT.

registers of on-chip peripheral modules are, however, initialized. For details on the state
chip peripheral module registers in software standby mode, refer to section 25.3, Registe
Each Operating Mode. For details on the pin states in software standby mode, refer to ap
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 th
2. Set the timer counter (WTCNT) of the WDT to 0 and bits CKS2 to CKS0 in WTCSI 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 (hig input to $\overline{\text{BREQ}}$ pin).
5. After setting the STBY bit in STBCR1 and the STBYMD bit in STBCR6 to 1, execu SLEEP instruction.
6. Software standby mode is entered and the clocks within this LSI are halted.

from being canceled.

When falling-edge detection is selected for the NMI pin, drive the NMI pin high before m transition to software standby mode. When rising-edge detection is selected for the NMI 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 making a transition to software standby mode. When rising-edge detection is selected for 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 rese the $\overline{\text{RES}}$ pin. Keep the $\overline{\text{RES}}$ pin low until the clock oscillation settles.

The contents of the CPU registers and the data of the on-chip RAM become undefined.
registers of on-chip peripheral modules are initialized. For details on the pin states in de
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 th
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 (hig input to $\overline{\text{BREQ}}$ pin).
4. After setting the STBY bit in STBCR1 to 1 and clearing the STBYMD bit in STBCF execute the SLEEP instruction.
5. Deep software standby mode is entered, the clocks within this LSI are halted, and th 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 th
low until the clock oscillation settles.

### 24.7.2 Canceling Module Standby Function

The module standby function can be canceled by clearing the MSTP bits in STBCR2 to S to 0. The module standby function can be canceled by a power-on reset for modules whos bit has an initial value of 0.

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- FRQCR twice before executing the SLEEP instruction.

- — 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 co 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 specific states such as reset, see the sections of those modules.

| Register Name | Abbreviation | No. of Bits | Address | Module | Size | Cycles |
|---|---|---|---|---|---|---|
| Serial mode register_0 | SCSMR_0 | 8 | H'FFFFC000 | SCI<br>(Channel 0) | 8 | Pϕ (reference clock)<br>B: 2 |
| 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<br>(Channel 1) | 8 | Pϕ (reference clock)<br>B: 2 |
| 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<br>(Channel 2) | 8 | Pϕ (reference clock)<br>B: 2 |
| 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 | |

| | | | | | |
|---|---|---|---|---|---|
| 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 |

| | | | | | |
|---|---|---|---|---|---|
| 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 |

| | | | | | |
|---|---|---|---|---|---|
| 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 |

| | | | | | |
|---|---|---|---|---|---|
| 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 |

|  |  |  |  |  |
|---|---|---|---|---|
| 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_4 S | 16 | H'FFFFC644 | 16, 32 |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 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)<br>B: 5 |
| 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 |  |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 | $P\phi$ reference<br>B: 2 |
| $I^2C$ bus control register 2 | ICCR2 | 8 | H'FFFFCD81 | | 8 | |
| $I^2C$ 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 | $P\phi$ (reference clock)<br>B: 2<br>W: 2 |
| SS control register L | SSCRL | 8 | H'FFFFCD01 | | 8 | |
| SS mode register | SSMR | 8 | H'FFFFCD02 | | 8, 16 | |
| 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 | $P\phi$ (reference clock)<br>B: 2<br>W: 2 |
| SS receive data register 0 | SSRDR0 | 8 | H'FFFFCD0A | | 8, 16 | |
| SS receive data register 1 | SSRDR1 | 8 | H'FFFFCD0B | | 8 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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)<br>B: 2<br>W: 2<br>L: 4 |
| Output level control/status register 1 | OCSR1 | 16 | H'FFFFD002 | | 8, 16 | |
| Input level control/status register 2 | ICSR2 | 16 | H'FFFFD004 | | 8, 16, 32 | |
| Output level control/status register 2 | OCSR2 | 16 | H'FFFFD006 | | 8, 16 | |
| 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)<br>B: 2<br>W: 2<br>L: 4 |
| Port A I/O register L | PAIORL | 16 | H'FFFFD106 | PFC | 8, 16 | |
| Port A control register L4 | PACRL4 | 16 | H'FFFFD110 | | 8, 16, 32 | |
| Port A control register L3 | PACRL3 | 16 | H'FFFFD112 | | 8, 16 | |
| 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 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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<br>(Channel 0) | 8 | Pϕ (reference clock)<br>B: 2<br>W: 2 |
| A/D status register_0 | ADSR_0 | 8 | H'FFFFD402 | | 8 | |
| A/D start trigger select register_0 | ADSTRGR_0 | 8 | H'FFFFD41C | | 8 | |
| 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 | |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 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)<br>B: 2<br>W: 2<br>L: 4 |
| 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,<br>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 |  |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  | 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 |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
|  | 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 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  | 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 |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
|  | 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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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)<br>W: 2 |
| Standby control register 1 | STBCR1 | 8 | H'FFFFE802 | Power-down modes | 8 | Pφ (reference clock)<br>B: 2 |
| 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 | 8*1, 16*2 | Pφ (reference clock) |
| 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 |
| Oscillation stop detection control register | OSCCR | 8 | H'FFFFE814 | CPG | 8 | Pφ (reference clock)<br>B: 2 |
| RAM control register | RAMCR | 8 | H'FFFFE880 | Power-down modes | 8 | Pφ (reference clock)<br>B: 2 |
| Bus function extending register | BSCEHR | 16 | H'FFFFE89A | BSC | 8, 16 | Pφ (reference clock)<br>B: 2<br>W: 2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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)<br>L: 2 |
| CS0 space bus control register | CS0BCR | 32 | H'FFFFF004 | | 32 | |
| 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 |
| Break address register A | BARA | 32 | H'FFFFF300 | UBC | 32 | Bϕ (reference clock)<br>B: 2<br>W: 2<br>L: 2 |
| Break address mask register A | BAMRA | 32 | H'FFFFF304 | | 32 | |
| Break bus cycle register A | BBRA | 16 | H'FFFFF308 | | 16 | |
| 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 | |

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 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] |  | S |
| SCBRR_1 |  |  |  |  |  |  |  |  | (C |
| 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] |  | S |
| SCBRR_2 |  |  |  |  |  |  |  |  | (C |
| 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 |  |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 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 | | | | | | | | |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| 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 | | | | | | | | |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 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] | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 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] | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 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] | | | MT |
| 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 | | | | | | | | | |
| | | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| 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 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 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 | | | | | | | | |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| TCNTCMPCLRS | — | — | — | — | — | CMPCLR5U | CMPCLR5V | CMPCLR5W |
| FCCS | FWE | MAT | — | FLER | — | — | — | SCO |
| FPCS | — | — | — | — | — | — | — | PPVS |
| FECS | — | — | — | — | — | — | — | EPVB |
| FKEY | K[7:0] | | | | | | | |
| FMATS | MS7 | MS6 | MS5 | MS4 | MS3 | MS2 | MS1 | MS0 |
| FTDAR | TDER | TDA[6:0] | | | | | | |
| DTCERA | DTCERA15 | DTCERA14 | DTCERA13 | DTCERA12 | — | — | — | — |
| | — | — | — | — | — | — | — | — |
| 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 | | | | | | | | |
| | | | | | | | | |
| | | | | | — | — | — | — |
| | — | — | — | — | — | — | — | — |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ICDRR | | | | | | | | | |
| NF2CYC | — | — | — | — | — | — | — | NF2CYC | |
| SSCRH | MSS | BIDE | — | SOL | SOLP | — | CSS[1:0] | | SS |
| 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 | — | — | — | — | — | — | — | — | CM |
| | — | — | — | — | — | — | STR1 | STR0 | |
| CMCSR_0 | — | — | — | — | — | — | — | — | |
| | CMF | CMIE | — | — | — | — | CKS[1:0] | | |
| CMCNT_0 | | | | | | | | | |
| | | | | | | | | | |
| CMCOR_0 | | | | | | | | | |
| | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | — | — | 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/ |
|  | PA7DR | PA6DR | PA5DR | PA4DR | PA3DR | PA2DR | PA1DR | PA0DR |  |
| PAIORL | PA15IOR | PA14IOR | PA13IOR | PA12IOR | PA11IOR | PA10IOR | PA9IOR | PA8IOR | P |
|  | 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 |  |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 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 | PF |
| | 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 | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 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/ |
| | — | — | PE21PR | PE20PR | PE19PR | PE18PR | PE17PR | PE16PR | |
| PEPRL | PE15PR | PE14PR | PE13PR | PE12PR | PE11PR | PE10PR | PE9PR | PE8PR | |
| | PE7PR | PE6PR | PE5PR | PE4PR | PE3PR | PE2PR | PE1PR | PE0PR | |
| IFCR | — | — | — | — | — | — | — | — | P |
| | — | — | — | — | — | — | IRQMD1 | IRQMD0 | |
| PFDRL | PF15DR | PF14DR | PF13DR | PF12DR | PF11DR | PF10DR | PF9DR | PF8DR | I/ |
| | PF7DR | PF6DR | PF5DR | PF4DR | PF3DR | PF2DR | PF1DR | PF0DR | |
| ADCR_0 | ADST | ADCS | ACE | ADIE | — | — | TRGE | EXTRG | A |
| 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] | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 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 |
| 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] | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 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,<br>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] | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | EXTID[7:0] | | | | | | | | |
| MB[0].<br>LAFMH | IDE_LAFM | — | — | STDID_LAFM[10:6] | | | | | RC |
| | STDID_LAFM[5:0] | | | | | | EXTID_LAFM[17:16] | | (MC |
| MB[0].<br>LAFMH | — | STDID_LAFM[10:4] | | | | | | | RC |
| | STDID_LAFM[3:0] | | | | — | IDE_LAFM | EXTID_LAFM[17:16] | | (MC |
| MB[0].<br>LAFML | EXTID_LAFM[15:8] | | | | | | | | RC |
| | EXTID_LAFM[7:0] | | | | | | | | |
| MB[0].<br>MSG_DATA[0] | MSG_DATA_0 | | | | | | | | |
| MB[0].<br>MSG_DATA[1] | MSG_DATA_1 | | | | | | | | |
| MB[0].<br>MSG_DATA[2] | MSG_DATA_2 | | | | | | | | |
| MB[0].<br>MSG_DATA[3] | MSG_DATA_3 | | | | | | | | |
| MB[0].<br>MSG_DATA[4] | MSG_DATA_4 | | | | | | | | |
| MB[0].<br>MSG_DATA[5] | MSG_DATA_5 | | | | | | | | |
| MB[0].<br>MSG_DATA[6] | MSG_DATA_6 | | | | | | | | |
| MB[0].<br>MSG_DATA[7] | MSG_DATA_7 | | | | | | | | |
| MB[0].<br>CONTROL1H | — | — | NMC | — | — | MBC[2:0] | | | |
| MB[0].<br>CONTROL1L | — | — | — | — | DLC[3:0] | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| FRQCR | — | IFC[2:0] | | | BFC[2:0] | | | PFC[2] |
| | PFC[1:0] | | MIFC[2:0] | | | MPFC[2:0] | | |
| STBCR1 | STBY | — | — | — | — | — | — | — |
| 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 | | | | | | | | |
| WTCSR | TME | WT/IT | RSTS | WOVF | IOVF | CKS[2:0] | | |
| OSCCR | — | — | — | — | — | OSCSTOP | — | OSCERS |
| RAMCR | — | — | — | RAME | — | — | — | — |
| BSCEHR | DTLOCK | CSSTP1 | — | CSSTP2 | DTBST | DTSA | CSSTP3 | DTPR |
| | — | — | — | — | — | — | — | — |
| ICR0 | NMIL | — | — | — | — | — | — | NMIE |
| | — | — | — | — | — | — | — | — |
| 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 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 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 | I$^2$C2 | I$^2$C2 | I$^2$C2 | I$^2$C2 |
| | RCAN-ET_0 | RCAN-ET_0 | RCAN-ET_0 | RCAN-ET_0 | — | — | — | — |
| CMNCR | — | — | — | — | — | — | — | — |
| | — | — | — | — | — | — | — | — |
| | — | — | — | — | — | — | — | — |
| | — | — | — | — | — | — | 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] | | — |
| | — | — | — | — | — | — | — | — |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | WR[0] | WM | — | — | — | — | HW[1:0] | |
| RAMER | — | — | — | — | — | — | — | — |
| | — | — | — | — | RAMS | RAM[2:0] | | |
| BARA | BAA31 | BAA30 | BAA29 | BAA28 | BAA27 | BAA26 | BAA25 | BAA24 |
| | 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 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 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 |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| SCSDCR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained |  |
| SCSPTR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained |  |
| SCSMR_1 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | SC<br>(C |
| 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 | SC<br>(C |
| 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 | M |
| 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 |  |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |
| TCR_4S | 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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 | $I^2C$ |
| 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 | SS |
| 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 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |
| 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 |
| PAIORL | Initialized | Retained | Retained | Initialized | — | Retained |
| 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 |
| PBDRL | Initialized | Retained | Retained | Initialized | — | Retained |
| PBIORL | Initialized | Retained | Retained | Initialized | — | Retained |
| PBCRL2 | Initialized | Retained | Retained | Initialized | — | Retained |
| PBCRL1 | Initialized | Retained | Retained | Initialized | — | Retained |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| PEDRH | Initialized | Retained | Retained | Initialized | — | Retained | |
| PEDRL | Initialized | Retained | Retained | Initialized | — | Retained | |
| PEIORH | Initialized | Retained | Retained | Initialized | — | Retained | PI |
| 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/C |
| PEPRL | Initialized | Retained | Retained | Initialized | — | Retained | |
| IFCR | Initialized | Retained | Retained | Initialized | — | Retained | PI |
| PFDRL | Initialized | Retained | Retained | Initialized | — | Retained | I/C |
| ADCR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | A/ |
| ADSR_0 | Initialized | Retained | Initialized | Initialized | Initialized | Retained | (C |
| 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 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |
| 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 |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| MSG_DATA[1] |  |  |  |  |  |  |  |
| MB[0].<br>MSG_DATA[2] | — | Retained | — | — | — | Retained |  |
| MB[0].<br>MSG_DATA[3] | — | Retained | — | — | — | Retained |  |
| MB[0].<br>MSG_DATA[4] | — | Retained | — | — | — | Retained |  |
| MB[0].<br>MSG_DATA[5] | — | Retained | — | — | — | Retained |  |
| MB[0].<br>MSG_DATA[6] | — | Retained | — | — | — | Retained |  |
| MB[0].<br>MSG_DATA[7] | — | Retained | — | — | — | Retained |  |
| MB[0].<br>CONTROL1H | Initialized | Retained | Initialized | Initialized | Initialized | Retained |  |
| MB[0].<br>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 | Cl |

| | | | | | | |
|---|---|---|---|---|---|---|
| WTCSR | Initialized* | Retained | Retained | Initialized | — | Retained |
| OSCCR | Initialized*[2] | Retained | Retained*[3] | Initialized | — | Retained |
| RAMCR | Initialized | Retained | Retained | Initialized | — | Retained |
| BSCEHR | Initialized | Retained | Retained | Initialized | — | Retained |
| ICR0 | Initialized | Initialized | Retained | Initialized | — | Retained |
| 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 |
| 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 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 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.

| Item | | Symbol | Value |
|---|---|---|---|
| Power supply voltage | | $V_{CC}$ | –0.3 to +7.0 |
| Input voltage (except analog input pins) | | $V_{in}$ | –0.3 to $V_{CC}$ +0.3 |
| Analog power supply voltage | | $AV_{CC}$ | –0.3 to +7.0 |
| Analog reference voltage | | $AV_{refh}$ | –0.3 to $AV_{CC}$ +0.3 |
| Analog input voltage | | $V_{an}$ | –0.3 to $AV_{CC}$ +0.3 |
| Operating temperature | Consumer applications | $T_{opr}$ | –20 to +85 |
| | Industrial applications | | –40 to +85 |
| Storage temperature | | $T_{stg}$ | –55 to +125 |

[Operating Precaution]

Operating the LSI in excess of the absolute maximum ratings may result in permanent da

| | | | | | | |
|---|---|---|---|---|---|---|
| Input high-level voltage (except Schmitt trigger input voltage) | $\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, FWE, MD1, MD0, $\overline{\text{ASEMD0}}$, 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) | $\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, FWE, MD1, MD0, $\overline{\text{ASEMD0}}$, EXTAL | $V_{IL}$ | –0.3 | — | 0.5 | V |
| | Other input pins | | –0.3 | — | 0.8 | V |
| Schmitt trigger input voltage | IRQ3 to IRQ0, $\overline{\text{POE8}}$, $\overline{\text{POE6}}$ to $\overline{\text{POE4}}$, $\overline{\text{POE2}}$ to $\overline{\text{POE0}}$ 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, $\overline{\text{SCS}}$, 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 $\overline{\text{ASEMD0}}$) | $\lvert I_{in} \rvert$ | — | — | 1.0 | μA |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | TIOC3BS, TIOC3DS,<br>TIOC4AS to TIOC4DS | | | | | | |
| | PE9, PE11 to PE21 | | $V_{CC}$–2.0 | — | — | V | $I_{OH}$ = |
| Output low-level voltage | All output pins | $V_{OL}$ | — | — | 0.4 | V | $I_{OL}$ = |
| | SCL, SDA | | — | — | 0.4 | V | $I_{OL}$ = |
| | | | — | — | 0.5 | V | $I_{OL}$ = |
| | TIOC3B, TIOC3D,<br>TIOC4A to TIOC4D,<br>TIOC3BS, TIOC3DS,<br>TIOC4AS to TIOC4DS | | — | — | 0.9 | V | $I_{OL}$ = |
| | PE9, PE11 to PE21 | | — | — | 2.0 | V | $I_{OL}$ = |
| Input capacitance | All input pins | $C_{in}$ | — | — | 20 | pF | $V_{in}$ =<br>f =<br>Ta = |
| Supply current | Normal operation | $I_{CC}$ | — | 80 | 105 | mA | Iϕ =<br>Bϕ =<br>Pϕ =<br>MPϕ<br>MIϕ |
| | Sleep | | — | 55 | 85 | mA | Bϕ =<br>Pϕ =<br>MPϕ<br>MIϕ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Standby | | — | — | 15 | μA | |
| Reference power supply current | During A/D conversion | $AI_{refh}$ | — | — | 2 | mA | The v A/D c modu |
| | 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 open.
2. The supply current was measured when $V_{IH}$ (Min.) = $V_{CC}$ – 0.5 V, $V_{IL}$ (Max.) = 0. all output pins unloaded.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Other input pins | | 2.2 | — | $V_{CC}$+0.3 | V | |
| Input low-level voltage (except Schmitt trigger input voltage) | $\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, FWE, MD1, MD0, $\overline{\text{ASEMD0}}$, EXTAL | $V_{IL}$ | –0.3 | — | 0.4 | V | |
| | Other input pins | | –0.3 | — | 0.8 | V | |
| Schmitt trigger input voltage | IRQ3 to IRQ0, $\overline{\text{POE8}}$, $\overline{\text{POE6}}$ to $\overline{\text{POE4}}$, $\overline{\text{POE2}}$ to $\overline{\text{POE0}}$ 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, $\overline{\text{SCS}}$, 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 $\overline{\text{ASEMD0}}$) | $\lvert I_{in} \rvert$ | — | — | 1.0 | μA | |
| Input pull-up MOS current | $\overline{\text{ASEMD0}}$ | $-I_{pu}$ | — | — | 800 | μA | $V_{in}$ = |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | PE9, PE11 to PE21 | | $V_{CC}$–2.0 | — | — | V | $I_{OH}$ = - |
| Output low-level voltage | All output pins | $V_{OL}$ | — | — | 0.4 | V | $I_{OL}$ = 1 |
| | SCL, SDA | | — | — | 0.4 | V | $I_{OL}$ = 3 |
| | | | — | — | 0.5 | V | $I_{OL}$ = 8 |
| | TIOC3B, TIOC3D,<br>TIOC4A to TIOC4D,<br>TIOC3BS, TIOC3DS,<br>TIOC4AS to TIOC4DS | | — | — | 1.4 | V | $I_{OL}$ = 1 |
| | PE9, PE11 to PE21 | | — | — | 1.5 | V | $I_{OL}$ = 1 |
| Input capacitance | All input pins | $C_{in}$ | — | — | 20 | pF | $V_{in}$ =<br>f = 1<br>Ta = |
| Supply current | Normal operation | $I_{CC}$ | — | 80 | 105 | mA | Iϕ = 8<br>Bϕ =<br>Pϕ =<br>MPϕ<br>MIϕ = |
| | Sleep | | — | 55 | 85 | mA | Bϕ =<br>Pϕ =<br>MPϕ<br>MIϕ = |
| | Software standby | | — | 8 | 20 | mA | $T_a$ ≤ 5 |
| | | | — | — | 30 | mA | 50°C |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| power supply current | conversion |  |  |  |  |  | A/D mod |
|  | 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 A open.
2. The supply current was measured when $V_{IH}$ (Min.) = $V_{CC}$ – 0.5 V, $V_{IL}$ (Max.) = all output pins unloaded.

| | | | | |
|---|---|---|---|---|
| Output high-level permissible current (total) | $\Sigma -I_{OH}$ | — | — | 35 |

[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 (Max.) about pins SCL and SDA. However, at most six pins are permitted to ha simultaneously $I_{OL}/-I_{OH} > 2.0$ mA among these pins.

| Item | | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| 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 | |

| | | | | | |
|---|---|---|---|---|---|
| EXTAL clock input frequency | $f_{EX}$ | 5 | 12.5 | MHz | Figure 2 |
| 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 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 2 |
| Standby return oscillation settling time 1 | $t_{OSC2}$ | 10 | — | ms | Figure 2 |
| Standby return oscillation settling time 2 | $t_{OSC3}$ | 10 | — | ms | Figure 2 |

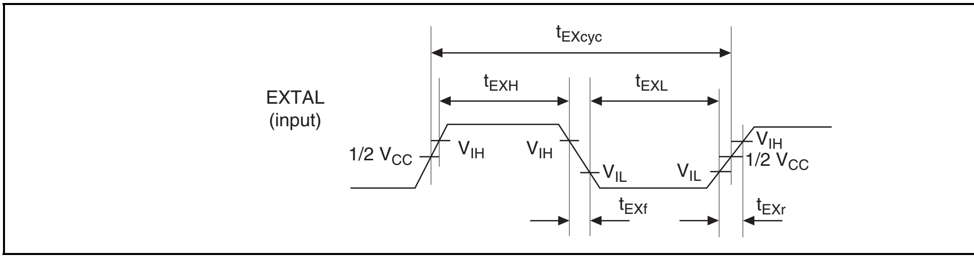
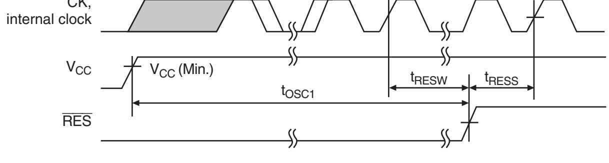
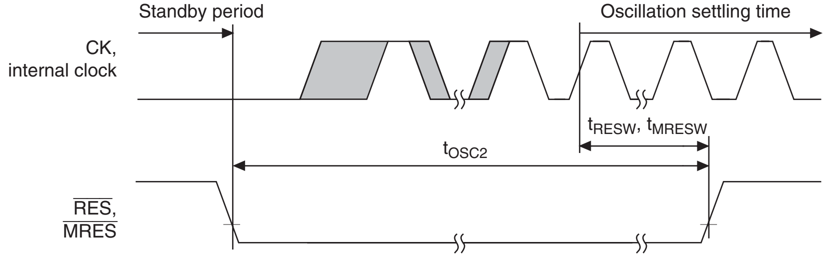


**Figure 26.1 EXTAL Clock Input Timing**



Note: Oscillation settling time when on-chip oscillator is used.

**Figure 26.3 Power-On Oscillation Settling Timing**



Note: Oscillation settling time when on-chip oscillator is used.

**Figure 26.4 Oscillation Settling Timing on Return from Standby (Return by**

Figure 26.5 Oscillation Setting Timing on Return from Standby (Return by NMI

| | | | | | |
|---|---|---|---|---|---|
| RES setup time*[1] | $t_{RESS}$ | 65 | — | ns | 26.6, 26.7 |
| $\overline{\text{RES}}$ hold time | $t_{RESH}$ | 15 | — | ns | |
| $\overline{\text{MRES}}$ pulse width | $t_{MRESW}$ | 20*[3] | — | $t_{Bcyc}$*[4] | |
| $\overline{\text{MRES}}$ setup time*[1] | $t_{MRESS}$ | 25 | — | ns | |
| $\overline{\text{MRES}}$ hold time | $t_{MRESH}$ | 15 | — | ns | |
| MD1, MD0, FWE setup time | $t_{MDS}$ | 20 | — | $t_{Bcyc}$*[4] | Figure 26 |
| $\overline{\text{BREQ}}$ setup time | $t_{BREQS}$ | $1/2t_{Bcyc} + 15$ | — | ns | Figure 26 |
| $\overline{\text{BREQ}}$ hold time | $t_{BREQH}$ | $1/2t_{Bcyc} + 10$ | — | ns | |
| NMI setup time*[1] | $t_{NMIS}$ | 60 | — | ns | Figure 26 |
| 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 | |
| $\overline{\text{IRQOUT}}$ output delay time | $t_{IRQOD}$ | — | 100 | ns | Figure 26 |
| $\overline{\text{BACK}}$ delay time | $t_{BACKD}$ | — | $1/2t_{Bcyc} + 20$ | ns | Figures 2 |
| Bus tri-state delay time | $t_{BOFF}$ | 0 | 100 | ns | |
| Bus buffer on time | $t_{BON}$ | 0 | 100 | ns | |

Notes: 1. The $\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, $\overline{\text{BREQ}}$, and IRQ3 to IRQ0 signals are asynchronous s When the setup time is satisfied, change of signal level is detected at the risin 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).

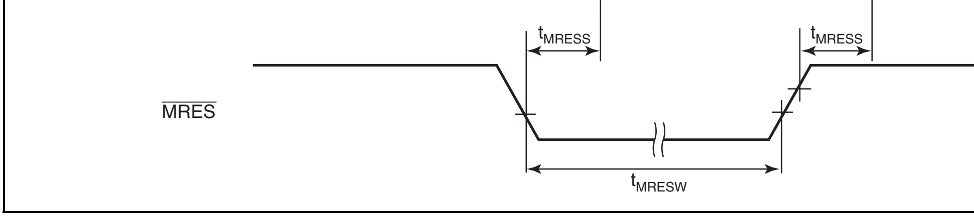


**Figure 26.6 Reset Input Timing**

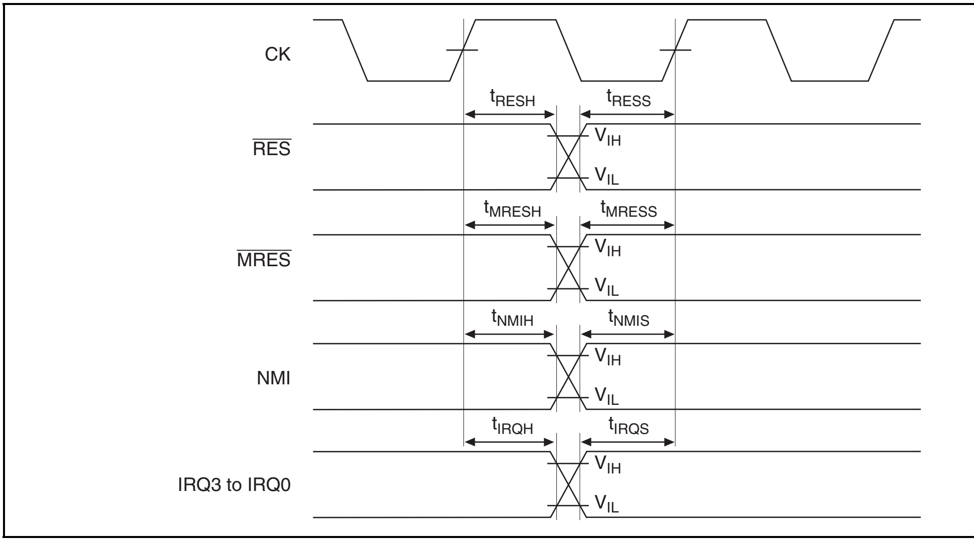


**Figure 26.7 Interrupt Signal Input Timing**

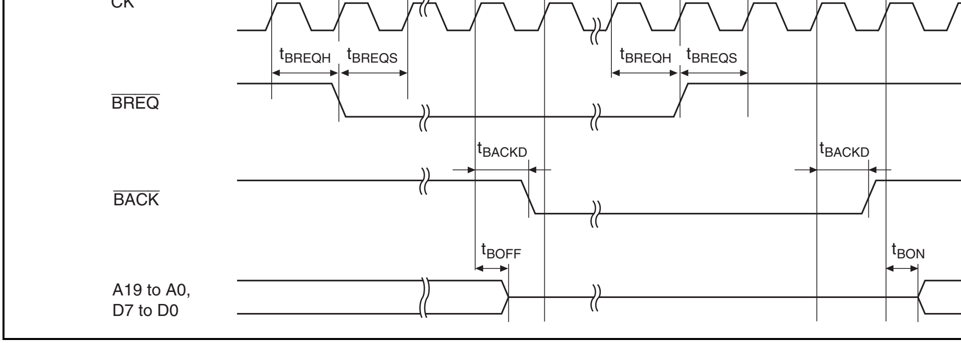


**Figure 26.9 Bus Release Timing**

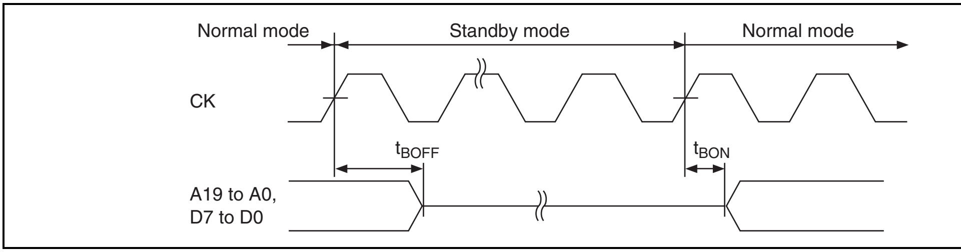


**Figure 26.10 Pin Driving Timing in Standby Mode**

| | | | | | |
|---|---|---|---|---|---|
| Address setup time | $t_{AS}$ | 0 | — | ns | Figures 2 26.14 |
| Address hold time | $t_{AH}$ | 0 | — | ns | Figures 2 26.14 |
| $\overline{\text{CS}}$ delay time | $t_{CSD}$ | 1 | 18 | ns | Figures 2 26.15 |
| $\overline{\text{CS}}$ setup time | $t_{CSS}$ | 0 | — | ns | Figures 2 26.14 |
| $\overline{\text{CS}}$ hold time | $t_{CSH}$ | 0 | — | ns | Figures 2 26.14 |
| Read strobe delay time | $t_{RSD}$ | $1/2t_{Bcyc}$ + 1 | $1/2t_{Bcyc}$ + 18 | ns | Figures 2 26.15 |
| Read data setup time 1 | $t_{RDS1}$ | $1/2t_{Bcyc}$ + 18 | — | ns | Figures 2 26.15 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Write data hold time 1 | $t_{WDH1}$ | 1 | 11 | ns | Figures<br>26.15 |
| Write data hold time | $t_{WRH}$ | 0 | — | ns | Figures<br>26.14 |
| $\overline{\text{WAIT}}$ setup time | $t_{WTS}$ | $1/2t_{Bcyc} + 17$ | — | ns | Figures<br>26.15 |
| $\overline{\text{WAIT}}$ hold time | $t_{WTH}$ | $1/2t_{Bcyc} + 7$ | — | ns | Figures<br>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 satisfied.

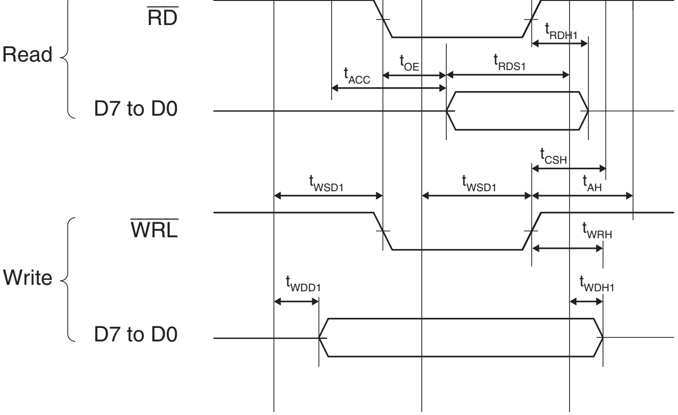


**Figure 26.11 Basic Bus Timing for Normal Space (No Wait)**

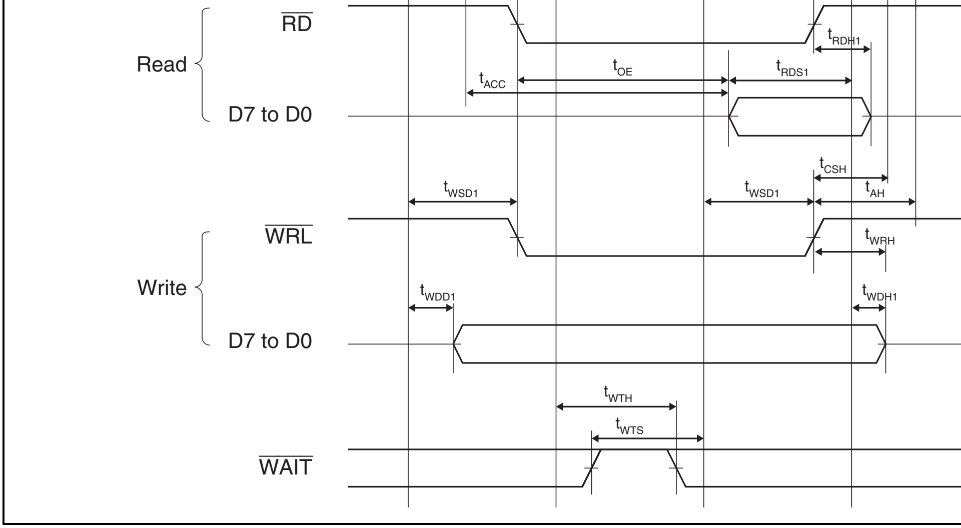


**Figure 26.12 Basic Bus Timing for Normal Space (One Software Wait Cyc**

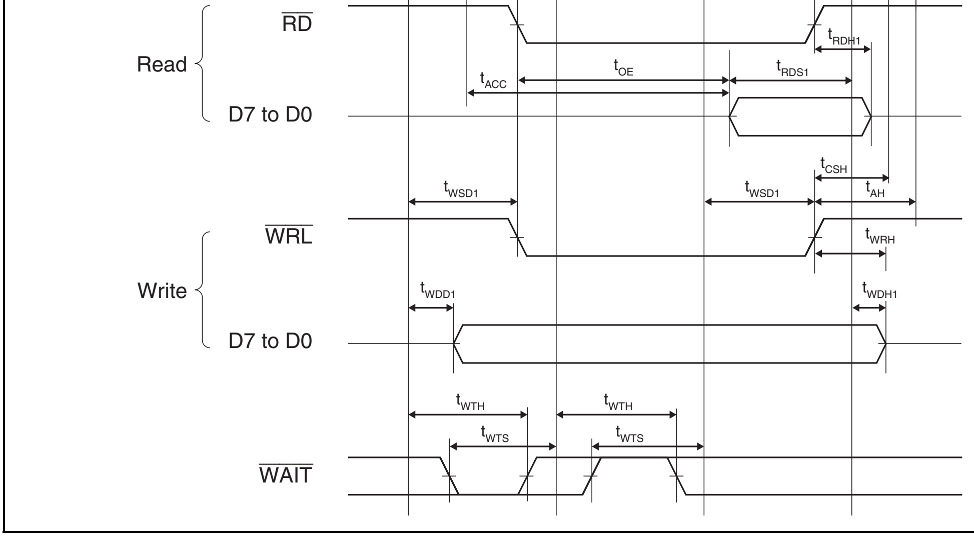


**Figure 26.13 Basic Bus Timing for Normal Space (One External Wait Cycl**

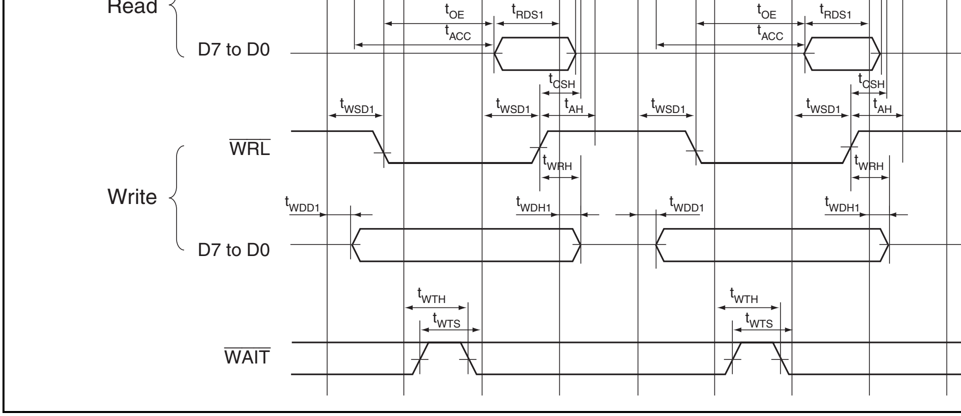


**Figure 26.14 Basic Bus Timing for Normal Space**
**(One Software Wait Cycle, External Wait Cycle Valid (WM Bit = 0), No Idle C**

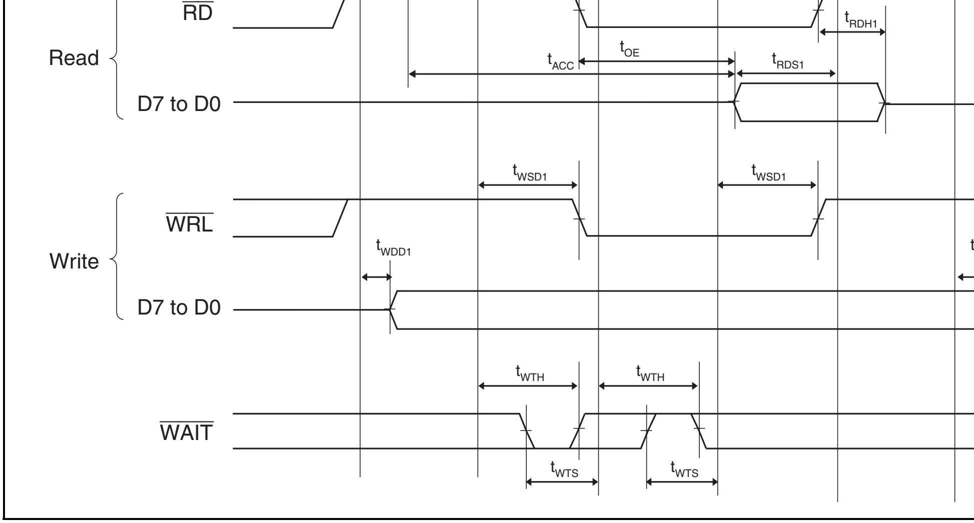


**Figure 26.15 CS Extended Bus Cycle for Normal Space (SW = 1 Cycle, HW = 1 Cycle, One External Wait Cycle)**

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Output compare output delay time | $t_{TOCD}$ |  | 50 | ns | Fig |
| 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 | Figu |
| 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.

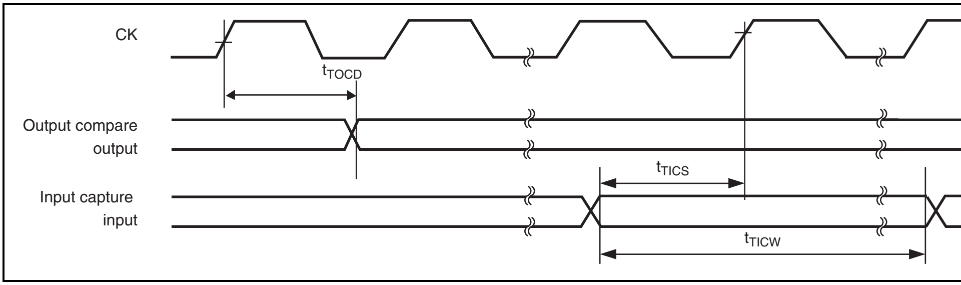
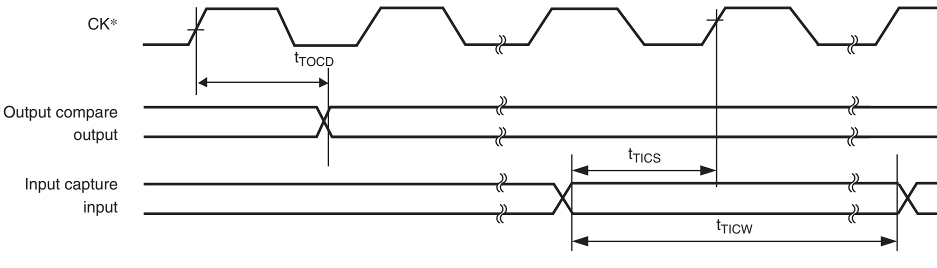


**Figure 26.16 MTU2 Input/Output Timing**

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Output compare output delay time | $t_{TOCD}$ |  | 50 | ns | Fig |
| 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.



Note: * When the MIϕ frequency is higher than the Bϕ frequency, MIϕ is used instead of CK.

**Figure 26.18 MTU2S Input/Output Timing**

| | | | | | |
|---|---|---|---|---|---|
| Port output data delay time | $t_{PWD}$ | | 30 | ns | Fig |
| Port input hold time | $t_{PRH}$ | 20 | — | ns | |
| Port input setup time | $t_{PRS}$ | 20 | — | ns | |

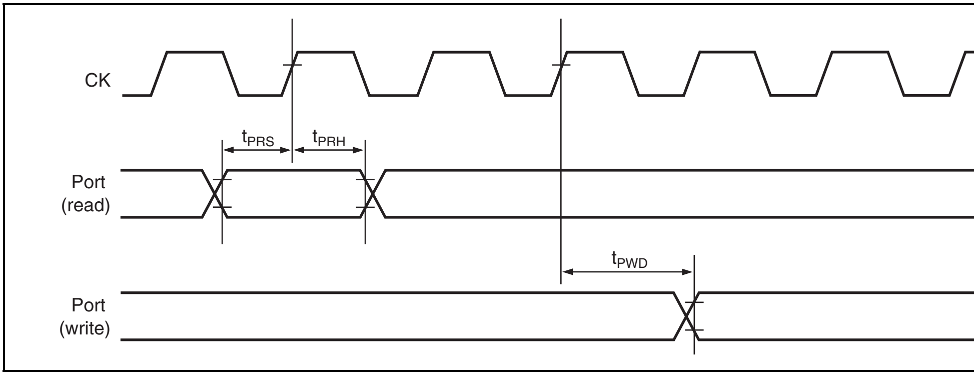


**Figure 26.19 I/O Port Input/Output Timing**

|  |  |  |  |  |
|---|---|---|---|---|
| WDTOVF delay time | $t_{WOVD}$ | 30 | ns | Fig |

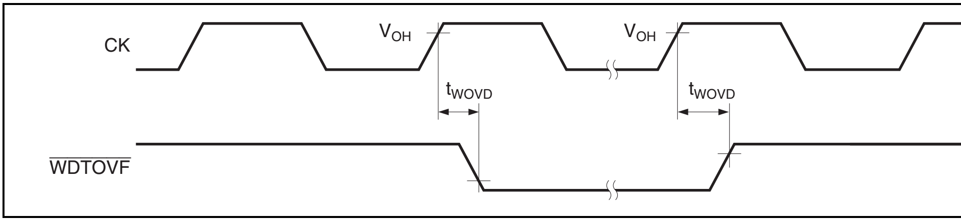


**Figure 26.20 WDT Timing**

| | | | | | | |
|---|---|---|---|---|---|---|
| Input clock cycle (asynchronous) | | $t_{scyc}$ | 4 | | $t_{pcyc}$ | F |
| Input clock cycle (clock synchronous) | | $t_{scyc}$ | 6 | — | $t_{pcyc}$ | 2 |
| 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 | F |
| Receive data setup time | | $t_{RXS}$ | 4 $t_{pcyc}$ | — | ns | 2 |
| 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.

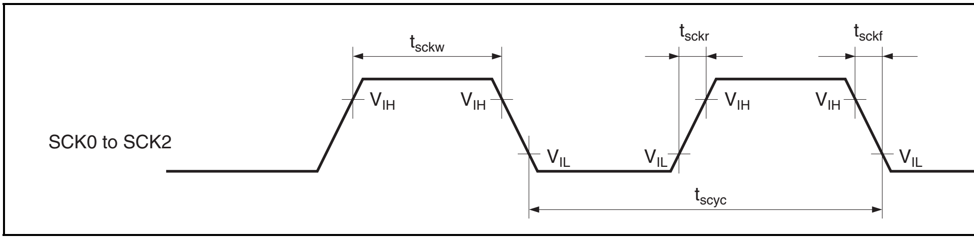


**Figure 26.21 Input Clock Timing**

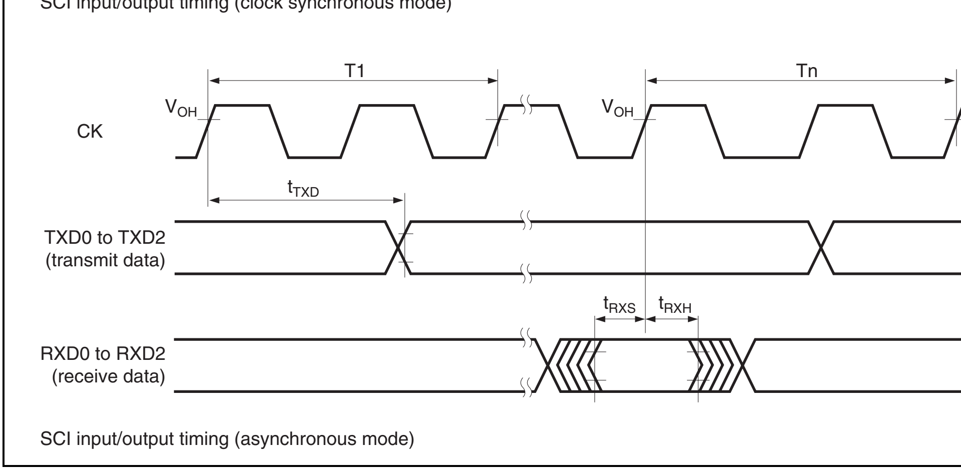

SCI input/output timing (clock synchronous mode)
T1
Tn
CK
VOH
VOH
tTXD
TXD0 to TXD2
(transmit data)
tRXS
tRXH
RXD0 to RXD2
(receive data)
SCI input/output timing (asynchronous mode)

**Figure 26.22 SCI Input/Output Timing**

| | | | | | | |
|---|---|---|---|---|---|---|
| | Slave | | 4 | 256 | | 26.26 |
| 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 | — | | |
| $\overline{\text{SCS}}$ setup time | Master | $t_{LEAD}$ | 1.5 | — | $t_{pcyc}$ | |
| | Slave | | 1.5 | — | | |
| $\overline{\text{SCS}}$ 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 26.26 |
| Slave out release time | | $t_{REL}$ | — | 1 | $t_{pcyc}$ | |

Note: $t_{pcyc}$ indicates the peripheral clock (Pϕ) cycle.

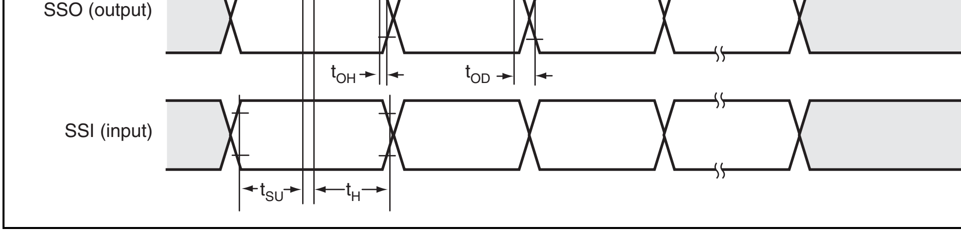


Figure 26.23 SSU Timing (Master, CPHS = 1)

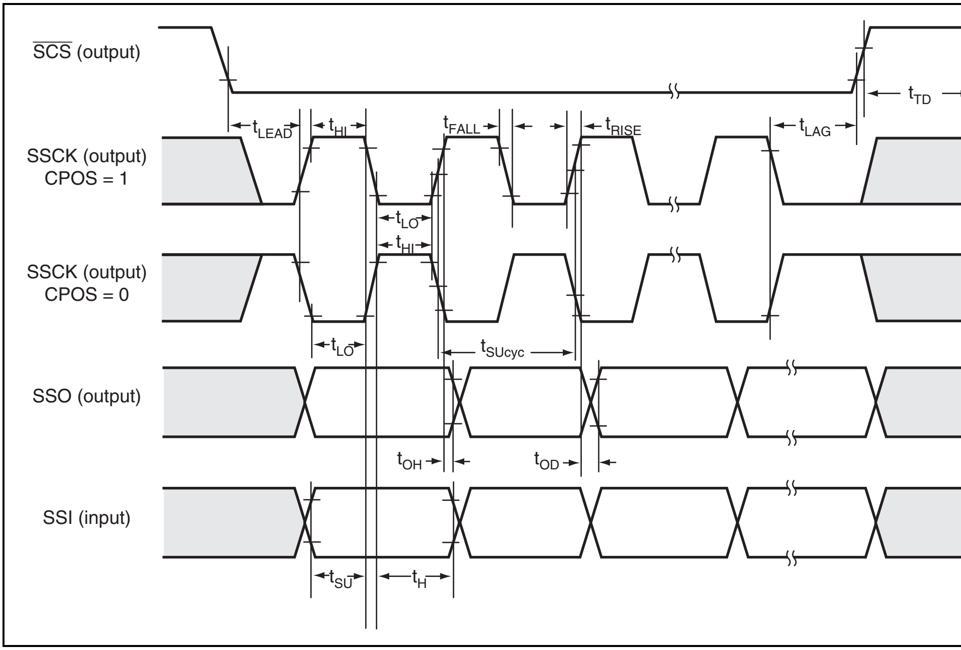


Figure 26.24 SSU Timing (Master, CPHS = 0)

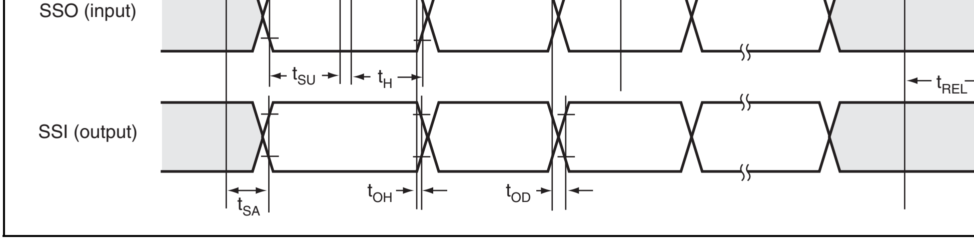


Figure 26.25 SSU Timing (Slave, CPHS = 1)

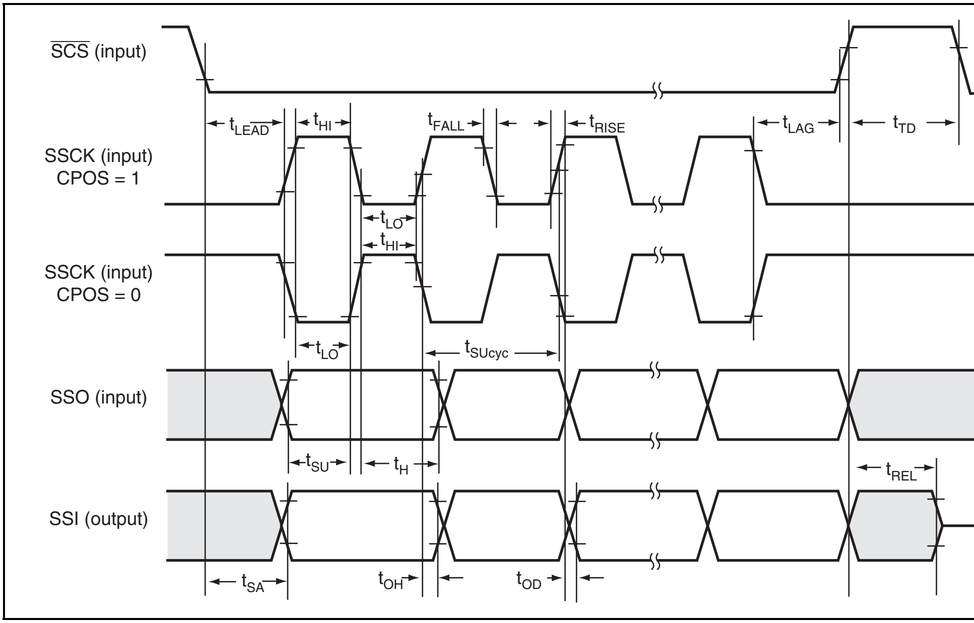


Figure 26.26 SSU Timing (Slave, CPHS = 0)

| Item | Symbol | Min. | Max. | Unit | Re<br>Fig |
|---|---|---|---|---|---|
| Transmit data delay time | $t_{CTxD}$ | — | 100 | ns | Fig |
| Receive data setup time | $t_{CRxS}$ | 100 | — | ns | |
| Receive data hold time | $t_{CRxH}$ | 100 | — | ns | |

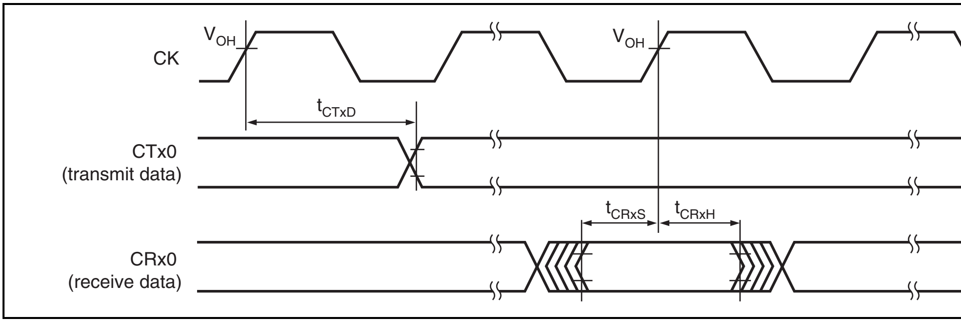


**Figure 26.27 RCAN-ET Input/Output Timing**

| | | | | | |
|---|---|---|---|---|---|
| $\overline{\text{POE}}$ input setup time | $t_{POES}$ | 50 | | ns | Figu |
| $\overline{\text{POE}}$ input pulse width | $t_{POEW}$ | 1.5 | — | $t_{pcyc}$ | |

Note: $t_{pcyc}$ indicates the peripheral clock (Pϕ) cycle.

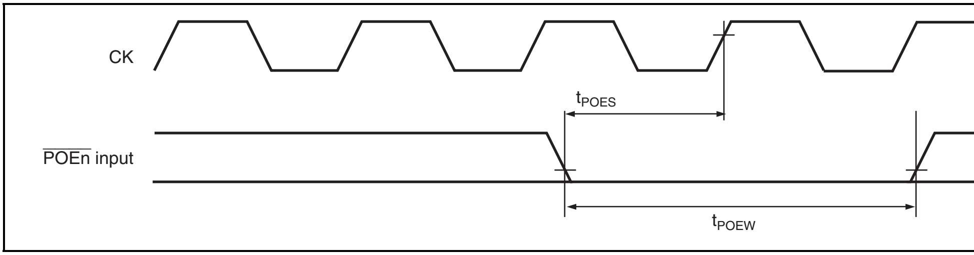


**Figure 26.28 $\overline{\text{POE}}$ Input Timing**

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| SCL input cycle time | $t_{SCL}$ | 12 $t_{pcyc}$ + 600 |  |  | ns | Figu |
| 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.

[Legend]
S: Start condition
P: Stop condition
Sr: Repeated start condition

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
$AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $AV_{SS}$ = $AV_{refl}$ = 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 | Refe<br>Figu |
|---|---|---|---|---|---|
| $\overline{\text{UBCTRG}}$ delay time | $t_{UBCTGD}$ | — | 150 | ns | Figu |

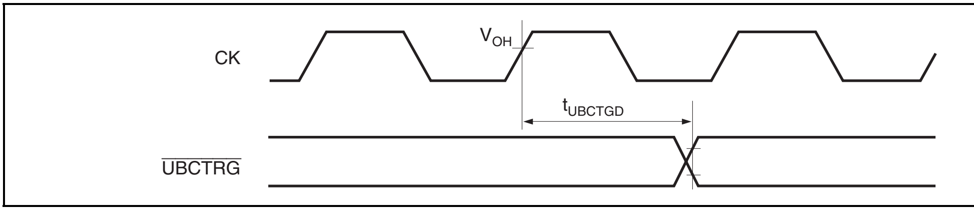


Figure 26.30 UBC Trigger Timing

delay time

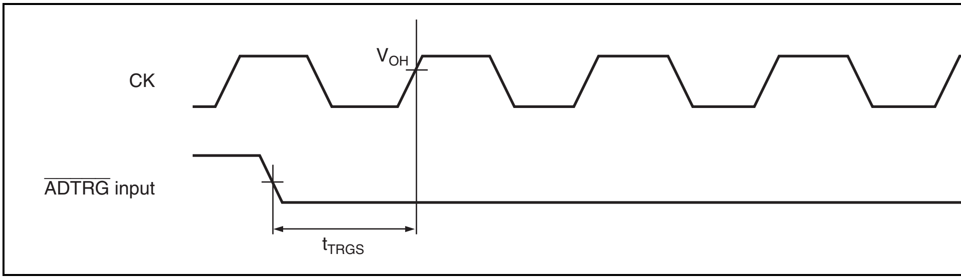
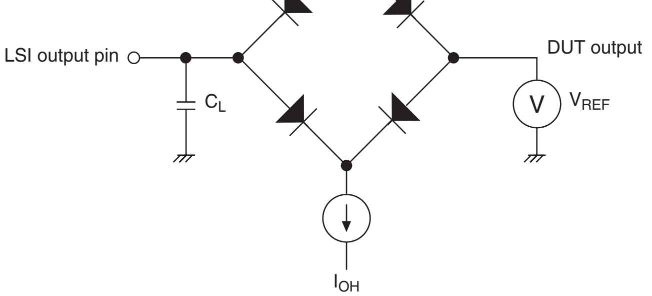


**Figure 26.31 External Trigger Input Timing**



Notes: 1. $C_L$ is the total value that includes the capacitance of measurement tools. Each pin is set as follo
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**

|  |  |  |  |
|---|---|---|---|
| A/D conversion time | 1.25*[1] | — | — |
| Analog input capacitance | — | — | 5 |
| Permitted analog signal source impedance | — | — | 3 |
| Non-linear error | — | — | ±4*[2] |
| Offset error | — | — | ±7.5*[2] |
| Full-scale error | — | — | ±7.5*[2] |
| Quantization error | — | — | ±0.5*[2] |
| Absolute error*[3] | — | — | ±8 |

Notes: 1. Conversion time per channel when the sample-and-hold circuit is not used ar clock operates at 40 MHz.

2. Reference value.

3. Guaranteed range from $AV_{refl}$ + 0.25 V to $AV_{refh}$ − 0.25 V.

| | | | | | |
|---|---|---|---|---|---|
| Erase time*1*2*4 | $t_E$ | — | 40 | 260 | ms/4 K<br>block |
| | | — | 300 | 1500 | ms/32<br>block |
| | | — | 600 | 3000 | ms/64<br>block |
| Programming time (total)*1*2*4 | $\Sigma t_P$ | — | 2.3 | 12 | s/256 |
| | | — | 1.1 | 6 | s/128 |
| Erase time (total)*1*2*4 | $\Sigma t_E$ | — | 2.3 | 12 | s/256 |
| | | — | 1.1 | 6 | s/128 |
| Programming and erase time (total)*1*2*4 | $\Sigma t_{PE}$ | — | 4.6 | 24 | s/256 |
| | | — | 2.2 | 12 | s/128 |
| Reprogramming count | $N_{WEC}$ | 500*3 | — | — | Times |

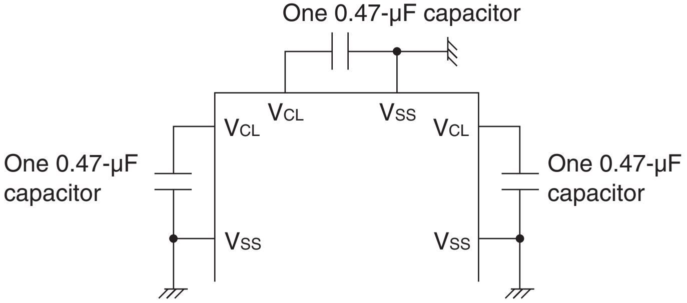
Notes: 1. Programming and erase time vary depending on the data.

2. Programming and erase time do not include data transfer time.

3. The minimum number of times for which all characteristics are guaranteed afte reprogramming (guaranteed for once to the minimum number of reprogrammin

4. These characteristics only apply when reprogramming is performed within the minimum number of times.

stabilizing capacitor



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**

| Type | Pin Name | Power-On | Manual | Deep Software Standby*[4] | Software Standby*[4] | Sleep | Oscillation Stop Detected |
|---|---|---|---|---|---|---|---|
| Clock | XTAL | O | O | L | L | O | O |
| | EXTAL | I | I | Z | I | I | I |
| System control | $\overline{\text{RES}}$ | I | I | I | I | I | I |
| | $\overline{\text{MRES}}$ | Z | I | Z | Z | I | Z |
| | $\overline{\text{WDTOVF}}$ | O*[2] | O | O | O | O | O |
| Operating mode control | MD1 | I | I | I | I | I | I |
| | $\overline{\text{ASEMD0}}$ | I*[3] | I*[3] | I*[3] | I*[3] | I*[3] | I*[3] |
| | FWE | I | I | I | I | I | I |
| Interrupt | NMI | I | I | I | I | I | I |
| | IRQ0 to IRQ3 | Z | I | Z | I | I | I |
| | $\overline{\text{IRQOUT}}$ | Z | O | Z | Z | O | Z |
| MTU2 | TCLKA to TCLKD | Z | I | Z | Z | I | I |
| | TIOC0A to TIOC0D | Z | I/O | Z | K*[1] | I/O | I/O |
| | TIOC1A, TIOC1B | Z | I/O | Z | K*[1] | I/O | I/O |
| | TIOC2A, TIOC2B | Z | I/O | Z | K*[1] | I/O | I/O |
| | TIOC3A, TIOC3C | Z | I/O | Z | K*[1] | I/O | I/O |
| | TIOC3B, TIOC3D | Z | I/O | Z | Z | I/O | Z |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | TIOC4DS | | | | | | | |
| | TIC5US, TIC5VS | Z | I | Z | Z | I | I | I |
| POE | $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$,<br>$\overline{\text{POE4}}$ to $\overline{\text{POE6}}$,<br>$\overline{\text{POE8}}$ | Z | I | Z | Z | I | I | I |
| SCI | SCK0 to SCK2 | Z | I/O | Z | Z | I/O | I/O | I/C |
| | 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/C |
| | $\overline{\text{SCS}}$ | Z | I/O | Z | Z | I/O | I/O | I/C |
| | SSI | Z | I/O | Z | Z | I/O | I/O | I/C |
| | SSO | Z | I/O | Z | Z | I/O | I/O | I/C |
| I$^2$C2 | SCL | Z | I/O | Z | Z | I/O | I/O | I/C |
| | SDA | Z | I/O | Z | Z | I/O | I/O | I/C |
| 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,<br>AN8 to AN15 | Z | I | Z | Z | I | I | I |
| | $\overline{\text{ADTRG}}$ | Z | I | Z | Z | I | I | I |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| PE11 to PE15 | | | | | | | |
| PE16 to PE21 | Z | I/O | Z | Z | I/O | Z | Z |
| PF0 to PF3,<br>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 regi (STBCR6) is set to 1.

2. Becomes input during a power-on reset. Pull-up to prevent erroneous operati 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.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 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 | I | Z |
| | $\overline{\text{WDTOVF}}$ | O*3 | | O | O | O | O | O | O |
| | $\overline{\text{BREQ}}$ | Z | | I | Z | Z | I | I | I |
| | $\overline{\text{BACK}}$ | Z | | O | Z | Z | O | L | O |
| Operating mode control | MD0, MD1 | I | | I | I | I | I | I | I |
| | $\overline{\text{ASEMD0}}$ | I*4 | | I*4 | I*4 | I*4 | I*4 | I*4 | I*4 |
| | 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 | O | Z |
| Address bus | A0 to A17 | O | Z | O | Z | Z*2 | O | Z | O |
| | A18, A19 | Z | | O | Z | Z*2 | O | Z | O |
| Data bus | D0 to D7 | Z | | I/O | Z | Z | I/O | Z | I/O |
| Bus control | $\overline{\text{WAIT}}$ | Z | | I | Z | Z | I | Z | I |
| | $\overline{\text{CS0}}$ (PE10) | H | Z | O | Z | Z*2 | O | Z | O |
| | $\overline{\text{CS0}}$ (PE17),<br>$\overline{\text{CS1}}$ (PE18) | Z | | O | Z | Z*2 | O | Z | O |
| | $\overline{\text{RD}}$ (PA6) | H | Z | O | Z | Z*2 | O | Z | O |
| | $\overline{\text{RD}}$ (PE19) | Z | | O | Z | Z*2 | O | Z | O |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | TIOC0A to TIOC0D | Z | I/O | Z | K*[1] | I/O | I/O | I/O |
| | 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 | I/O | Z |
| | TIOC4A to TIOC4D | Z | I/O | Z | Z | I/O | I/O | Z |
| | TIC5U, TIC5V, TIC5W | Z | I | Z | Z | I | I | I |
| MTU2S | TIOC3BS, TIOC3DS | Z | I/O | Z | Z | I/O | I/O | Z |
| | TIOC4AS to TIOC4DS | Z | I/O | Z | Z | I/O | I/O | Z |
| | TIC5US, TIC5VS, TIC5WS | Z | I | Z | Z | I | I | I |
| POE | $\overline{\text{POE0}}$ to $\overline{\text{POE2}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, $\overline{\text{POE8}}$ (PA9) | Z | I | Z | Z | I | I | I |

|  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
| SSU | SSCK | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O |
|  | SCS (overlined) | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O |
|  | SSI | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O |
|  | SSO | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O |
| I$^2$C2 | SCL | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O |
|  | SDA | Z | Z | Z | I/O | Z | Z | I/O | I/O | I/O |
| UBC | $\overline{\text{UBCTRG}}$*5 | Z |  |  | O | Z | O*1 | O | O | O |
| RCAN-ET | CTx0 | Z | Z | O*1 | O | O | O | O | RCAN-ET | CTx0 |
|  | CRx0 | Z | Z | Z | I | I | I | I |  | CRx0 |
| A/D Converter | AN0 to AN15 | Z |  |  | I | Z | Z | I | I | I |
|  | $\overline{\text{ADTRG}}$ | Z |  |  | I | Z | Z | I | I | I |
| I/O Port | PA0 to PA15 | Z |  |  | I/O | Z | K*1 | I/O | I/O | I/O |
|  | PB0 to PB7 | Z |  |  | I/O | Z | K*1 | I/O | I/O | I/O |
|  | PD0 to PD10 | 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 | I/O |
|  | 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 | I/O | Z |
|  | PE16 to PE21 | Z |  |  | I/O | Z | Z | I/O | I/O | Z |
|  | PF0 to PF15 | Z |  |  | I | Z | Z | I | I | I |

3. Becomes input during a power-on reset. Pull-up to prevent erroneous operati 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 resis 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) 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 th emulator.

| | | | | |
|---|---|---|---|---|
| | 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)**

| | | External Space (Normal Space) |
|---|---|---|
| **Pin Name** | | **8-bit Space** |
| $\overline{\text{CS0}}$, $\overline{\text{CS1}}$ | | Enabled |
| $\overline{\text{RD}}$ | R | L |
| | W | H |
| $\overline{\text{WRL}}$ | R | H |
| | W | L |
| A19 to A0 | | Address |
| D7 to D0 | | Data |

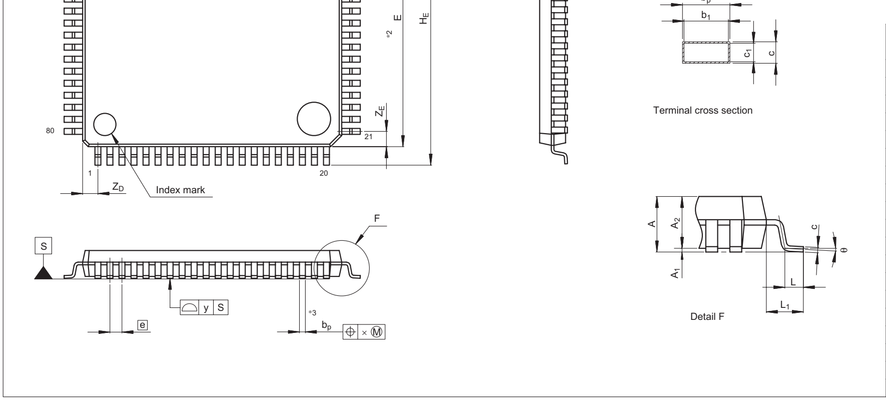
[Legend]

R: Read

W: Write

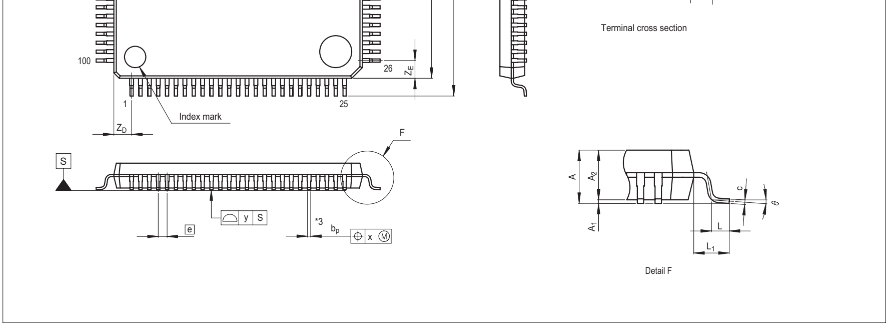
Enabled: Chip select signals corresponding to accessed areas = Low.
The other chip select signals = High.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 256 Kbytes | 16 Kbytes | | | R5F71324AD80FPV | |
| SH7136 | F-ZTAT version | 256 Kbytes | 16 Kbytes | Consumer application | −20 to +85°C | R5F71364AN80FPV | LQ<br>(FP |
| | | | | Industrial application | −40 to +85°C | R5F71364AD80FPV | |
| SH7137 | F-ZTAT version | 256 Kbytes | 16 Kbytes | Consumer application | −20 to +85°C | R5F71374AN80FPV | LQ<br>(FP |
| | | | | Industrial application | −40 to +85°C | R5F71374AD80FPV | |



| Reference Symbol | Dime |
|---|---|
| | Mi |
| D | 13. |
| E | 13. |
| $A_2$ | — |
| $H_D$ | 15. |
| $H_E$ | 15. |
| A | — |
| $A_1$ | 0.0 |
| $b_p$ | 0.2 |
| $b_1$ | — |
| c | 0.0 |
| $c_1$ | — |
| θ | 0 |
| e | — |
| x | — |
| y | — |
| $Z_D$ | — |
| $Z_E$ | — |
| L | 0.3 |
| $L_1$ | — |

**Figure E.1 FP-80WV**



| | |
|---|---|
| E | 1 |
| $A_2$ | - |
| $H_D$ | 1 |
| $H_E$ | 1 |
| A | - |
| $A_1$ | 0 |
| $b_p$ | 0 |
| $b_1$ | - |
| c | 0 |
| $c_1$ | - |
| $\theta$ | |
| e | - |
| x | - |
| y | - |
| $Z_D$ | - |
| $Z_E$ | - |
| L | 0 |
| $L_1$ | - |

**Figure E.2 FP-100UV**

SH7136, and SH7137

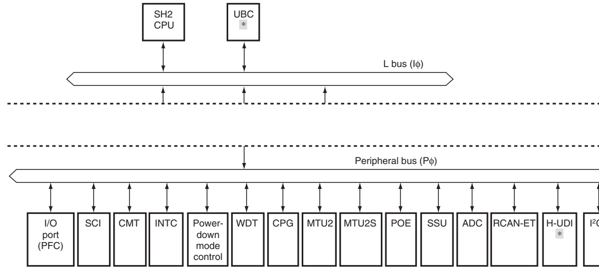
| | |
|---|---|
| Operating modes | • Operating modes<br>Single chip mode<br>Extended ROM enabled mode (Only in SH7132/SH7<br>Extended ROM disabled mode (Only in SH7132/SH7<br>• Operating states<br>Program execution state<br>Exception handling state<br>Bus release state (Only in SH7132/SH7137)<br>• Power-down modes<br>Sleep mode<br>Software standby mode (Only in SH7136/SH7137)<br>Deep software standby mode (Only in SH7136/SH71<br>Module standby mode |
| User break controller (UBC) | • Addresses, data values, type of access, and data size ca break conditions<br>• Supports a sequential break function |
| (SH7132 and SH7137 only) | • Two break channels |

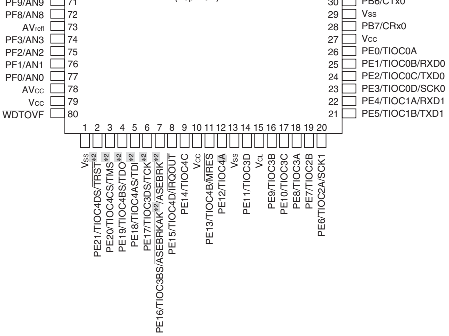
3

| 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 are CS1) (Only in SH7132/SH7137)<br>• 8-bit external bus (Only in SH7132/SH7137)<br>• The following features settable for each area independe<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

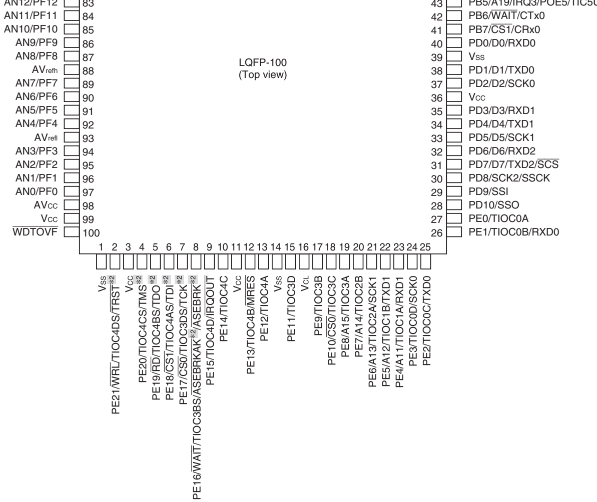
| Items | Specification |
|---|---|
| User debugging interface (H-UDI) (SH7136 and SH7137 only) | • Supports the E10A emulator |

| | | |
|---|---|---|
| | | • LQFP1414-80 (0.65 pitch) (SH7131/SH7136) |
| 1.2 Block Diagram<br>Figure 1.1 Block Diagram | 7 | Figure amended and note added<br><br>Note: * SH7136 and SH7137 only. |



Notes:

1. This pin is used by the E10A emulator. On the SH71
   functions as a Vcc fixed pin and on the SH7136 as t
   ASEMD0 input pin.
2. Pin function enabled on the SH7136 only.



Notes:

1. This pin is used by the E10A emulator. On the SH713 functions as a Vcc fixed pin and on the SH7137 as th $\overline{\text{ASEMD0}}$ input pin.

2. Pin function enabled on the SH7137 only.

<table>
<tr><td></td><td></td><td>

| | $\overline{\text{TRST}}$ | I | Test reset | Initialization-signal inp |
|---|---|---|---|---|

</td></tr>
<tr><td></td><td>16</td><td>

| Classification | Symbol | I/O | Name | Function |
|---|---|---|---|---|
| E10A interface (SH7136 and SH7137 only) | $\overline{\text{ASEMDO}}$ | I | ASE mode | Sets the ASE mode. driven low, the LSI er mode, and when driv operates in normal m dedicated functions c the ASE mode. Wher input to this pin, it is p internally. |
| | $\overline{\text{ASEBRK}}$ | I | Break request | E10A emulator break |
| | $\overline{\text{ASEBRKAK}}$ | O | Break mode acknowledge | Indicates the E10A er entered the break mc |

</td></tr>
<tr><td>3.4 Address Map<br>Figure 3.1 Address Map for Each Operating Mode in SH7131 (128-Kbyte Flash Memory Version)</td><td>52</td><td>Figure added</td></tr>
<tr><td>Figure 3.2 Address Map for Each Operating Mode in SH7131 and SH7136 (256-Kbyte Flash Memory Version)</td><td>53</td><td>Figure title amended</td></tr>
<tr><td>Figure 3.3 Address Map for Each Operating Mode in SH7132 (128-Kbyte Flash Memory Version)</td><td>54</td><td>Figure added</td></tr>
<tr><td>Figure 3.4 Address Map for Each Operating Mode in SH7132 and SH7137 (256-Kbyte Flash Memory Version)</td><td>55</td><td>Figure title amended</td></tr>
</table>

Table 5.1 Types of Exceptions and Priority

| | |
|---|---|
| Interrupt | User break (break after instruction execution or operan |

Notes: 3. SH7136 and SH7137 only.

5.1.3 Exception Handling Vector Table

Table 5.3 Vector Numbers and Vector Table Address Offsets

79, 80

Table amended and note amended

| Exception Handling Source | | Vector Number | Vector Table Address |
|---|---|---|---|
| Interrupt | NMI | 11 | H'0000002C to H'00000 |
| | User break*1 | 12 | H'00000030 to H'00000 |

| Exception Handling Source | Vector Number | Vector Table Address |
|---|---|---|
| On-chip peripheral module*2 | 72 | H'00000120 to H'00000 |
| | : | : |
| | 255 | H'000003FC to H'00000 |

Notes:

1. SH7136 and SH7137 only.
2. For details on the vector numbers and vector table ac offsets of on-chip peripheral module interrupts, see ta in section 6, Interrupt Controller (INTC).

5.4.1 Interrupt Sources

Table 5.7 Interrupt Sources

85

Table amended and note added

| Type | Request Source | Nu So |
|---|---|---|
| User break* | User break controller (UBC) | 1 |

Note: * SH7136 and SH7137 only.

5.4.2 Interrupt Priority

Table 5.8 Interrupt Priority

86

Table amended and note added

| Type | Priority Level | Comment |
|---|---|---|
| User break* | 15 | Fixed priority level. Can be maske |

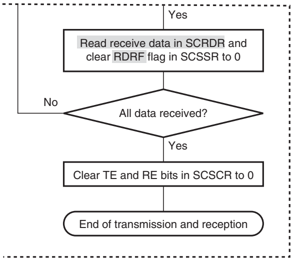
Note: * SH7136 and SH7137 only.

<table>
<tr><td></td><td></td><td>

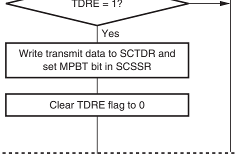
Note: * SH7136 and SH7137 only.</td></tr>
<tr><td>6.4.3 User Break Interrupt (SH7136 and SH7137 only)</td><td>109</td><td>Title amended</td></tr>
<tr><td>6.5 Interrupt Exception Handling Vector Table<br>Table 6.3 Interrupt Exception Handling Vectors and Priorities</td><td>110</td><td>Table amended and note amended
<table>
<tr><th>Interrupt Source</th><th>Name</th><th>Vector No.</th><th>Vector Table Starting Address</th><th>IP</th></tr>
<tr><td>User break*1</td><td></td><td>12</td><td>H'00000030</td><td></td></tr>
</table></td></tr>
<tr><td></td><td>113</td><td>Notes:<br>1. SH7136 and SH7137 only.<br>2. Of the $\text{I}^2$C2 interrupts, the vector address for the IIN interrupt is separated from others.</td></tr>
<tr><td>Section 7 User Break Controller (UBC) (SH7136 and SH7137 only)</td><td>123</td><td>Title amended</td></tr>
</table>

| | | |
|---|---|---|
| Flash Memory Version) in Single-Chip Mode | | |
| 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 |

| | | |
|---|---|---|
| | | For the Tb interval at the trough in comp PWM mode, see figure 10.40.<br>0: Outputs the initial value specified in T<br>1: Suppresses initial output<br>[Setting condition]<br>• When 1 is written to WRE after readi |
| 10.4.8 Complementary PWM Mode | 384 | Description amended<br>14. …Even in the Tb interval at the trough, if synchrono clearing occurs in the initial value output period (indicat in figure 10.56) immediately after the counters start ope initial value output is not suppressed.<br>When using the initial output suppression function, mak set compare registers TGRB_3, TGRA_4, and TGRB_ value twice or more the setting of dead time data regist If synchronous clearing occurs with the compare regist a value less than twice the setting of TDDR, the PWM dead time may be too short (or nonexistent) or illegal a PWM negative-phase output may occur during the initia suppression interval. For details, see 10.7.23, Notes or Waveform Control During Synchronous Counter Cleari Complementary PWM Mode. |

<table>
<tr><td>Bits T3AEN and T4VEN in Timer Interrupt Skipping Set Register (TITCR) and Buffer Transfer-Enabled Period</td><td></td><td></td></tr>
<tr><td>10.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode</td><td>453, 454</td><td>Newly added</td></tr>
<tr><td>13.5 Interrupt Source</td><td>532</td><td>Newly added</td></tr>
<tr><td>14.3.8 Serial Port Register (SCSPTR)</td><td>550</td><td>Table amended
<table>
<tr><td>Bit:</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>EIO</td><td>-</td><td>-</td><td>-</td><td>SPB1IO</td><td>SPB1DT</td><td>SPB0IO</td><td>SPB0DT</td></tr>
<tr><td>Initial value:</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>-</td><td>0</td><td>1</td></tr>
<tr><td>R/W:</td><td>R/W</td><td>-</td><td>-</td><td>-</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table></td></tr>
<tr><td>14.4.3 Clock Synchronous Mode<br>Figure 14.16 Sample Flowchart for Transmitting/Receiving Serial Data</td><td>583</td><td>Figure amended

Yes
Read receive data in SCRDR and clear RDRF flag in SCSSR to 0
No
All data received?
Yes
Clear TE and RE bits in SCSCR to 0
End of transmission and reception

</td></tr>
</table>



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.

14.5 SCI Interrupt Sources and DTC

590

Description amended

When the ORER, FER, or PER flag in SCSSR is set to interrupt request is generated. This request cannot be u activate the DTC. It is possible to disable generation of interrupt requests and allow only ERI interrupt requests generated during data reception processing. To accom set the RIE bit to 1 and the EIO bit in SCSPTR to 1. No setting the EIO bit to 1 will prevent the DTC from transf received data because no ERI interrupt requests are ge

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 $\overline{\text{SC}}$ output.<br>00: Setting prohibited<br>01: Setting prohibited<br>10: Function as $\overline{\text{SCS}}$ automatic input/ou $\overline{\text{SCS}}$ input before and after transfer a low level during transfer)<br>11: Function as $\overline{\text{SCS}}$ automatic output (c level before and after transfer and o level during transfer) |

| Bit | Bit Name | Value | R/W | Description |
|---|---|---|---|---|
| 2 | TDRE | 1 | R/W | Transmit Data Empty<br>Indicates whether or not SSTDR contains<br>[Setting conditions]<br>• When the TE bit in SSER is 0<br>• When data is transferred from SSTDR and SSTDR is ready to be written to.<br>[Clearing conditions]<br>• When writing 0 after reading TDRE =<br>• When writing data to SSTDR with TE =<br>• When the DTC is activated by an SST and transmit data is written to SSTDR DISEL bit in MRB of the DTC is 0 (exc DTC transfer counter value is H'0000) |

DTC transfer counter value is H'000

| | | | | |
|---|---|---|---|---|
| 0 | CE | 0 | R/W | Conflict/Incomplete Error |

Indicates that a conflict error has occurre
when 0 is externally input to the SCS pin
= 0 (SSU mode) and MSS = 1 (master m

If the SCS pin level changes to 1 with SS
mode) and MSS = 0 (slave mode), an in
occurs because it is determined that a m
has terminated the transfer. In addition,
0 (SSU mode) and MSS = 0 (slave mod
serial receive operation starts while RDF
incomplete error occurs even if the data
SSRDR is read before the completion of
RDRF is cleared to 0 before the SCS pin

Data reception does not continue while t
to 1. Serial transmission also does not c
the SSU internal sequencer by setting th
SSCRL to 1 before resuming transfer aft
error.

[Setting conditions]

- When a low level is input to the SCS
mode (the MSS bit in SSCRH is set
- When the SCS pin is changed to 1 d
slave mode (the MSS bit in SSCRH i
- When in slave mode (MSS = 0 in SS
serial receive operation starts while R
data is read from SSRDR before the
reception, after which the SCS pin is

[Clearing condition]

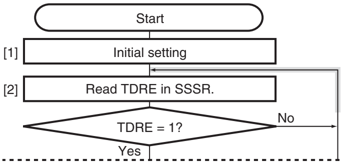
- When writing 0 after reading CE = 1

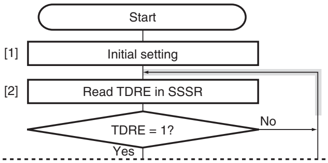
15.4.4 Communication Modes and Pin Functions 618

Table 15.7 Communication Modes and Pin States of SCS Pin

Table amended

| Communication Mode | Register Setting | | | |
|---|---|---|---|---|
| | SSUMS | MSS | CSS1 | CSS0 |
| SSU communication mode | 0 | 0 | x | x |
| | | 1 | 0 | 0 |
| | | | 0 | 1 |

| | | |
|---|---|---|
| | | that an overrun error (OEI) has occurred. At this time, d<br>reception is stopped. While the ORER bit in SSSR is set<br>reception is not performed. To resume the reception, cle<br>ORER bit to 0.<br>When setting the SSU to slave mode to perform continu<br>reception, read SSRDR before starting the next receive<br>operation. If the next receive operation starts before SSI<br>read and RDRF is cleared to 0, and SSRDR is read befo<br>reception completes, CE in SSSR is set to 1 after the co<br>of reception.<br>In addition, if the next receive operation starts before SS<br>read and RDRF is cleared to 0, and SSRDR is not read<br>reception completes, the receive data is discarded even<br>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 receptio<br>(RE = 1) to transmission/reception mode (TE = RE = 1),<br>TE and RE bits to 0. When starting the transfer, confirm<br>TEND, RDRF, and ORER bits are cleared to 0 before se<br>TE or RE bit to 1.<br>If the value of RDRF is 1 when the 8th clock rises, OREI<br>SSSR is set to 1, an overrun error occurs, and reception<br>Receive operation is not possible while ORER is set to 1<br>restart reception, first clear ORER to 0. |
| Figure 15.9 Flowchart Example of Simultaneous Transmission/Reception (SSU Mode) | | Figure amended<br><br> Start<br>[1]<br>Initial setting<br>[2]<br>Read TDRE in SSSR.<br>TDRE = 1?<br>No<br>Yes<br>[1] Initial setting:<br>Specify the  transmit/receive data forr<br>[2] Check the SSU state and write transr<br>Write transmit data to SSTDR after re<br>confirming that the TDRE bit in SSSR<br>bit is automatically cleared to 0 and tr<br>reception is started by writing data to |

| | | |
|---|---|---|
| 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> Start<br>[1]<br>Initial setting<br>[2]<br>Read TDRE in SSSR<br>TDRE = 1?<br>No<br>Yes<br>[1] Initial setting:<br>Specify the transmit data format.<br>[2] Check that the SSU state and write trans<br>Write transmit data to SSTDR after readi<br>that the TDRE bit is 1. The TDRE bit is a<br>to 0 and transmission is started by writin |
| (3) Data Reception | 633 | Description amended<br>When 1-frame data has been received, the RDRF bit i set to 1 and the receive data is stored in SSRDR. At th the RIE bit is set to 1, an RXI interrupt is generated. Th bit is automatically cleared to 0 by reading SSRDR.<br>When setting the SSU to slave mode to perform contin reception, read SSRDR before starting the next receive operation. If the next receive operation starts before S read and RDRF is cleared to 0, the integrity of subseq cannot be guaranteed. |
| (4) Data Transmission/Reception | 635 | Description added<br>Before switching transmission mode (TE = 1) or recept (RE = 1) to transmission/reception mode (TE = RE = 1 TE and RE bits to 0. When starting the transfer, confir TEND, RDRF, and ORER bits are cleared to 0 before TE or RE bits to 1.<br>If the value of RDRF is 1 when the 8th clock rises, ORE SSSR is set to 1, an overrun error occurs, and receptic Receive operation is not possible while ORER is set to restart reception, first clear ORER to 0. |

| | | |
|---|---|---|
| Reception Operations in SSU Slave Mode | | |
| 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$^2$C 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$^2$C occupied or released and to issue start/sto in master mode. With the clock synchrono format, this bit is always read as 0. With th format, this bit is set to 1 when the SDA le from high to low under the condition of SC assuming that the start condition has been bit is cleared to 0 when the SDA level cha to high under the condition of SCL = high, that the stop condition has been issued. T condition, simultaneously write 1 to BBSY SCP. Follow this procedure also when tra repeated start condition. To issue a stop c simultaneously write 0 to BBSY and 0 to S |
| 6 | SCP | 1 | R/W | Start/Stop Issue Condition Disable<br>The SCP bit controls the issue of start/stop master mode.<br>To issue a start condition, simultaneously BBSY and 0 to SCP. A repeated start cond issued in the same way. To issue a stop c simultaneously write 0 to BBSY and 0 to S is always read as 1. Even if 1 is written to data will not be stored. |

| | | |
|---|---|---|
| | 684 | Bit \| Bit Name \| Value \| R/W \| Description<br>4 \| NAKIE \| 0 \| R/W \| NACK Receive Interrupt Enable<br>NAKIE enables or disables the NACK detection/arbitration lost/overrun error in (IINAKI) when the NACKF or AL/OVE b to 1. IINAKI can be canceled by clearing AL/OVE, or NAKIE bit to 0.<br>0: NACK receive interrupt request (IINA<br>1: NACK receive interrupt request (IINA |
| 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 a 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 |

3. It takes about one bit time quantum for the value to r

4. If there is no TXPR set, RCAN-ET will receive the ne incoming message. If there is a TXPR(s) set, RCAN- start transmission of the message and will be arbitrat CAN bus. If it loses the arbitration, it will become a re

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) | Functior (Related |
|---|---|---|---|---|
| A | PA6 I/O (port) | $\overline{\text{UBCTRG}}$ output (UBC)* | TCLKA input (MTU2) | $\overline{\text{POE4}}$ inpu |

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) | Fun (Rel |
|---|---|---|---|---|---|
| A | PA6 I/O (port) | $\overline{\text{RD}}$ output (BSC) | $\overline{\text{UBCTRG}}$ output (UBC)* | TCLKA input (MTU2) | $\overline{\text{POE}}$ |

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 (Related M |
|---|---|---|---|---|
| E | PE16 I/O (port) | TIOC3BS I/O (MTU2S) | $\overline{\text{ASEBRKAK}}$ output (E10A)* | $\overline{\text{ASEBRK}}$ i |
| | 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.

(MTU2S)

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 | ASEMD0*[1] | ASEMD0*[1] |
| 45 | PA6 | PA6/UBCTRG*[2]/TCLKA/POE4 |

| | Pin Name | |
|---|---|---|
| | Single-Chip Mode (MCU Mode 3) | |
| Pin No. | Initial Function | Functions Selectable by PFC |
| 7 | PE16/(ASEBRKAK/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/(TRST*[1]) | P E21/TIOC4DS |

794

Notes:

1. Fixed to TMS, TRST, TDI, TDO, TCK, and ASEBRKAK/ASEBRK when using the E10A (ASEM

2. Function enabled on the SH7136 only.

| | | |
|---|---|---|
| 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 ($\overline{\text{ASEMD}}$
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 | |
| Pin No. | Initial Function | Functions Selectable by PFC | Initial Function | Functions S<br>PFC |
| 62 | PA6 | PA6/$\overline{\text{RD}}$/$\overline{\text{UBCTRG}}$*2/TCLKA/<br>$\overline{\text{POE4}}$ | PA6 | PA6/$\overline{\text{UBCTRG}}$ |

| | Pin Name | | | |
|---|---|---|---|---|
| | On-Chip ROM Enabled (MCU Mode 2) | | Single-Chip Mode (MCU | |
| Pin No. | Initial Function | Functions Selectable by PFC | Initial Function | Functions S<br>PFC |
| 8 | PE16/($\overline{\text{ASEBRKAK}}$/<br>$\overline{\text{ASEBRK}}$*1) | PE16/$\overline{\text{WAIT}}$/TIOC3BS | PE16/($\overline{\text{ASEBRKAK}}$/<br>$\overline{\text{ASEBRK}}$*1) | PE16/TIOC3B |
| 7 | PE17/(TCK*1) | PE17/$\overline{\text{CS0}}$/TIOC3DS | PE17/(TCK*1) | PE17/TIOC3D |
| 6 | PE18/(TDI*1) | PE18/$\overline{\text{CS1}}$/TIOC4AS | PE18/(TDI*1) | PE18/TIOC4A |
| 5 | PE19/(TDO*1) | PE19/$\overline{\text{RD}}$/TIOC4BS | PE19/(TDO*1) | PE19/TIOC4B |
| 4 | PE20/(TMS*1) | PE20/TIOC4CS | PE20/(TMS*1) | PE20/TIOC4C |
| 2 | PE21/($\overline{\text{TRST}}$*1) | PE21/$\overline{\text{WRL}}$/TIOC4DS | PE21/($\overline{\text{TRST}}$*1) | PE21/TIOC4D |

802 Notes:

1. Fixed to TMS, $\overline{\text{TRST}}$, TDI, TDO, TCK, and $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$ when using the E10A ($\overline{\text{ASEMD}}$
2. Function enabled on the SH7137 only.

setting on the SH7131.

SH7132/SH7137:

- 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/$\overline{RD}$/$\overline{UBCTRG}$/TCLKA/$\overline{POE4}$ pin. |
| 8 | PA6MD0 | 0*[1] | R/W | 000: PA6 I/O (port)<br>001: TCLKA input (MTU2)<br>011: $\overline{RD}$ output (BSC)*[2]<br>101: $\overline{UBCTRG}$ output (UBC)*[3]<br>111: $\overline{POE4}$ input (POE)<br>Other than above: Setting prohibited |

SH7132/SH7137:

- Port A Control Register L2 (PACRL2)

816

Notes:

1. The initial value is 1 in the on-chip ROM disabled ex extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not s value in single-chip mode.
3. Function enabled on the SH7137 only. Do not use t on the SH7132.

| 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/IRQ3/PO |
| 4 | PB5MD0 | 0 | R/W | 000: PB5 I/O (port)<br>001: IRQ3 input (INTC)<br>011: TIC5U input (MTU2)<br>111: 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/POI pin. |
| 0 | PB4MD0 | 0 | R/W | 000: PB4 I/O (port)<br>001: IRQ2 input (INTC)<br>011: TIC5US input (MTU2S)<br>111: POE4 input (POE)<br>Other than above: Setting prohibited |

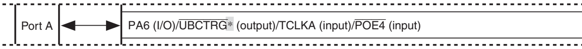
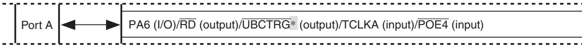
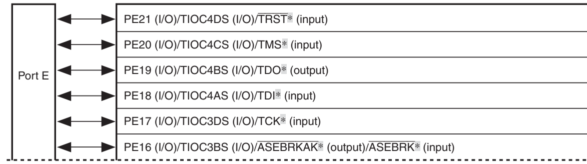
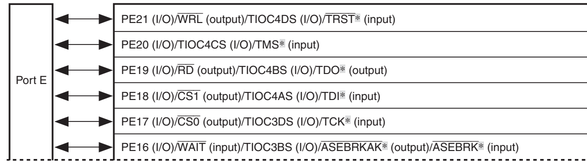
SH7132/SH7137:

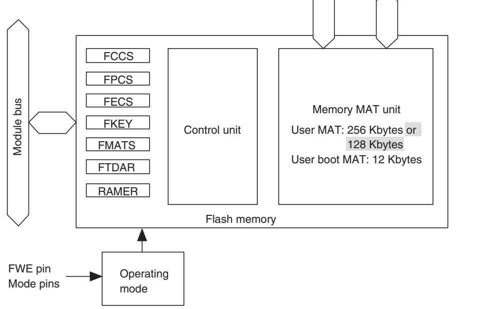
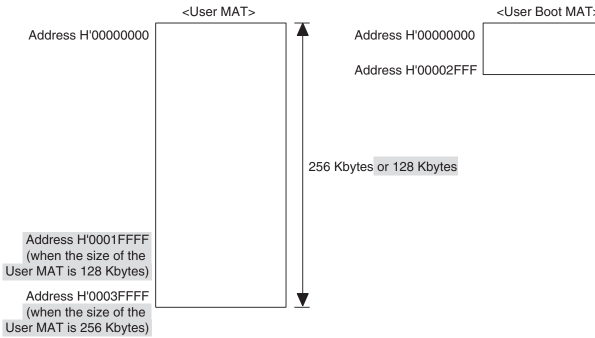
- Port B Control Register L2 (PBCRL2)

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/CS1/CRx |
| 12 | PB7MD0 | 0 | R/W | 000: PB7 I/O (port)<br>101: 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/WAIT/CT |
| 8 | PB6MD0 | 0 | R/W | 000: PB6 I/O (port)<br>101: WAIT input (BSC)*<br>110: CTx0 output (RCAN-ET)<br>Other than above: Setting prohibited |

| | | |
|---|---|---|
| 21.1 Port A<br>Figure 21.1 Port A (SH7131/SH7136) | 854 | Figure amended and note added<br><br> Port A<br>PA6 (I/O)/$\overline{UBCTRG}$* (output)/TCLKA (input)/$\overline{POE4}$ (input)<br>Note: * SH7136 only. |
| Figure 21.2 Port A (SH7132/SH7137) | 855 | Figure amended and note added<br><br> Port A<br>PA6 (I/O)/$\overline{RD}$ (output)/$\overline{UBCTRG}$* (output)/TCLKA (input)/$\overline{POE4}$ (input)<br>Note: * SH7137 only. |
| 21.4 Port E<br>Figure 21.6 Port E (SH7131/SH7136) | 869 | Figure amended and note added<br><br> Port E<br>PE21 (I/O)/TIOC4DS (I/O)/$\overline{TRST}$* (input)<br>PE20 (I/O)/TIOC4CS (I/O)/TMS* (input)<br>PE19 (I/O)/TIOC4BS (I/O)/TDO* (output)<br>PE18 (I/O)/TIOC4AS (I/O)/TDI* (input)<br>PE17 (I/O)/TIOC3DS (I/O)/TCK* (input)<br>PE16 (I/O)/TIOC3BS (I/O)/$\overline{ASEBRKAK}$* (output)/$\overline{ASEBRK}$* (input)<br>Note: * SH7136 only. |
| Figure 21.7 Port E (SH7132/SH7137) | 870 | Figure amended and note added<br><br> Port E<br>PE21 (I/O)/$\overline{WRL}$ (output)/TIOC4DS (I/O)/$\overline{TRST}$* (input)<br>PE20 (I/O)/TIOC4CS (I/O)/TMS* (input)<br>PE19 (I/O)/$\overline{RD}$ (output)/TIOC4BS (I/O)/TDO* (output)<br>PE18 (I/O)/$\overline{CS1}$ (output)/TIOC4AS (I/O)/TDI* (input)<br>PE17 (I/O)/$\overline{CS0}$ (output)/TIOC3DS (I/O)/TCK* (input)<br>PE16 (I/O)/$\overline{WAIT}$ (input)/TIOC3BS (I/O)/$\overline{ASEBRKAK}$* (output)/$\overline{ASEBRK}$* (input)<br>Note: * SH7137 only. |
| Section 22 Flash Memory | 881 | Description amended<br>This LSI has 256-Kbyte or 128-Kbyte on-chip flash mer flash memory has the following features. |

|  |  |  |
|---|---|---|
| Memory |  |  Module bus<br>FCCS<br>FPCS<br>FECS<br>FKEY<br>FMATS<br>FTDAR<br>RAMER<br>Control unit<br>Memory MAT unit<br>User MAT: 256 Kbytes or 128 Kbytes<br>User boot MAT: 12 Kbytes<br>Flash memory<br>FWE pin<br>Mode pins<br>Operating mode |
| 22.2.4 Flash Memory Configuration | 887 | Description amended<br>This LSI's flash memory is configured by the 256-Kbyte Kbyte user MAT and 12-Kbyte user boot MAT. |
| Figure 22.3 Flash Memory Configuration |  | Figure amended<br> <User MAT><br>Address H'00000000<br>Address H'0001FFFF (when the size of the User MAT is 128 Kbytes)<br>Address H'0003FFFF (when the size of the User MAT is 256 Kbytes)<br>256 Kbytes or 128 Kbytes<br><User Boot MAT:<br>Address H'00000000<br>Address H'00002FFF |
| 22.2.5 Block Division | 888 | Description amended<br>The user MAT is divided into 64 Kbytes (256-Kbyte vers blocks, 128-Kbyte version: one block), 32 Kbytes (one b and 4 Kbytes (eight blocks) as shown in figure 22.4. |

22.4.3 Programming/Erasing Interface Parameters

913

Table amended

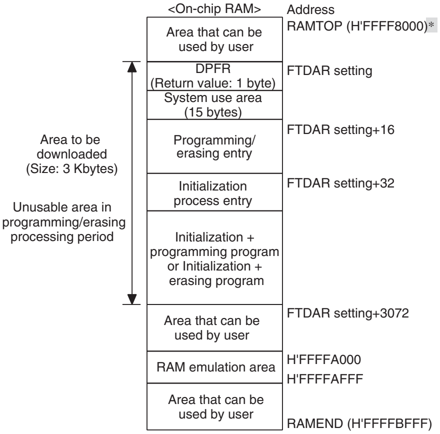
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | EBS[7:0] | Undefined | R/W | • 256-Kbyte flash memory<br>Set the erase-block number in the ra<br>11. 0 corresponds to the EB0 block a<br>corresponds to the EB11 block. An e<br>when a number other than 0 to 11 (H<br>set.<br>• 128-Kbyte flash memory<br>Set the erase-block number in the ra<br>0 corresponds to the EB0 block and<br>to the EB9 block. An error occurs wh<br>other than 0 to 9 (H'00 to H'09) is se |

22.5.2 User Program Mode

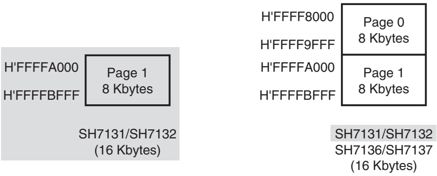
Figure 22.10 RAM Map after Download

923

Figure amended and note added


<On-chip RAM>
Address
RAMTOP (H'FFFF8000)*
Area that can be used by user
DPFR (Return value: 1 byte)
FTDAR setting
System use area (15 bytes)
Programming/ erasing entry
FTDAR setting+16
Initialization process entry
FTDAR setting+32
Area to be downloaded (Size: 3 Kbytes)
Unusable area in programming/erasing processing period
Initialization + programming program or Initialization + erasing program
FTDAR setting+3072
Area that can be used by user
RAM emulation area
H'FFFFA000
H'FFFFAFFF
Area that can be used by user
RAMEND (H'FFFFBFFF)

Note: * H'FFFFA000 on SH7131/SH7132.

| | | |
|---|---|---|
| 22.8.3 Other Notes | 948 | Description added<br>5. Note on programming the product having a 128-Kbyt MAT<br>If an attempt is made to program the product having Kbyte user MAT with more than 128 Kbytes, data programmed after the first 128 Kbytes are not guarar |
| 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 co at 4) |
| 22.10 Programmer Mode | 986 | Description amended<br>Use a PROM programmer that supports the Renesas 12 256-Kbyte flash memory on-chip MCU device type (F-Z |
| Section 23 RAM | 987 | Figure amended<br> H'FFFFA000<br>Page 1<br>8 Kbytes<br>H'FFFFBFFF<br>SH7131/SH7132<br>(16 Kbytes)<br>H'FFFF8000<br>Page 0<br>8 Kbytes<br>H'FFFF9FFF<br>H'FFFFA000<br>Page 1<br>8 Kbytes<br>H'FFFFBFFF<br>SH7131/SH7132<br>SH7136/SH7137<br>(16 Kbytes) |
| 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 |

|  |  |  |
|---|---|---|
|  |  | 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 Acce |
|---|---|---|---|---|---|---|
| Watchdog timer counter | WTCNT | 8 | H'FFFFE810 | WDT | 8*1, 16*2 | Pϕ (referenc |
| 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/ |
|---|---|---|---|---|---|---|---|---|
| IPRM | SSU | SSU | SSU | SSU | I$^2$C2 | I$^2$C2 | I$^2$C2 | I$^2$ |
|  | RCAN-ET_0 | RCAN-ET_0 | RCAN-ET_0 | RCAN-ET_0 | — | — | — |  |

| Item | | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | | | — | — | 0.5 | V |
| | TIOC3B, TIOC3D,<br>TIOC4A to TIOC4D,<br>TIOC3BS, TIOC3DS,<br>TIOC4AS to TIOC4DS | | — | — | 0.9 | V |
| | PE9, PE11 to PE21 | | — | — | 2.0 | V |

Table 26.3 DC Characteristics

1068

Table amended

| Item | | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Output high-level voltage | All output pins | $V_{OH}$ | $V_{CC}$–0.5 | — | — | V |
| | | | $V_{CC}$–1.0 | — | — | V |
| | TIOC3B, TIOC3D,<br>TIOC4A to TIOC4D,<br>TIOC3BS, TIOC3DS,<br>TIOC4AS to TIOC4DS | | $V_{CC}$–1.0 | — | — | V |
| | PE9, PE11 to PE21 | | $V_{CC}$–2.0 | — | — | V |
| Output low-level voltage | All output pins | $V_{OL}$ | — | — | 0.4 | V |
| | SCL, SDA | | — | — | 0.4 | V |
| | | | — | — | 0.5 | V |
| | TIOC3B, TIOC3D,<br>TIOC4A to TIOC4D,<br>TIOC3BS, TIOC3DS,<br>TIOC4AS to TIOC4DS | | — | — | 1.4 | V |
| | PE9, PE11 to PE21 | | — | — | 1.5 | V |

| | | | | |
|---|---|---|---|---|
| Programming and erase time (total)*1*2*4 | $\Sigma t_{PE}$ | — | 4.6 | 24 |
| | | — | 2.2 | 12 |
| Reprogramming count | $N_{WEC}$ | 500*3 | — | — |

A. Pin States

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 | Oscillatio Stop Dete |
| UBC | $\overline{\text{UBCTRG}}$*4 | Z | O | Z | O*1 | 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

|  |  |  |  |  |
|---|---|---|---|---|
|  |  |  | application |  |
|  |  |  | 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 |  |  |

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

Changing frequency ........................
Clock (MIϕ) for the MTU2S modu
Clock (MPϕ) for the MTU2 modu
Clock frequency control circuit....
Clock operating mode ..................
Clock pulse generator (CPG) .......
Clock synchronous mode ............
Clock synchronous serial format
(I$^2$C2)............................................
Clock timing ...............................
CMT interrupt sources ................
Compare match timer (CMT) ......
Complementary PWM mode .......
Conflict between NMI interrupt ar
DTC activation..............................
Connecting crystal resonator........
Continuous scan mode .................
Control signal timing ...................
Controller area network (RCAN-E
CPU...............................................
Crystal oscillator ..........................
$\overline{\text{CSn}}$ assert period extension .........

**D**

Data transfer controller (DTC).....
Data transfer instructions .............
DC characteristics ........................
Dead time compensation..............
Deep software standby mode .......

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

Initiation intervals of user branch processing ......................................
Input sampling and A/D conversion time ..............................................
Instruction formats..........................
Instruction set.................................
Interrupt controller (INTC)............
Interrupt exception handling vector table..............................................
Interrupt priority ............................
Interrupt response time ..................
Interrupt sequence..........................
Interrupts........................................
IRQ interrupts ................................

L
List of registers ..............................
Local acceptance filter mask (LAF
Location of transfer information and DTC vector table....................
Logic operation instructions ..........

M
Mailbox..........................................
Mailbox control..............................
Mailbox structure...........................
Manual reset...................................
MCU extension mode ....................
MCU operating modes...................

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

Pin states of bus related signals....
Pin states of this LSI in each processing state ............................
Port output enable (POE) ............
Power-down modes......................
Power-down state.........................
Power-on reset .............................
Procedure register (PR)................
Product code lineup......................
Program counter (PC) ..................
Program execution state ..............
Programmer mode.........................

Q
Quantization error ........................

R
RAM .............................................
RCAN-ET bit rate calculation......
RCAN-ET interrupt sources.........
RCAN-ET memory map ..............
RCAN-ET reset sequence ............
Reconfiguration of Mailbox.........
Register
ABACK0.................................
ADANSR_0 and ADANSR_1 .
ADCR_0 and ADCR_1...........
ADDR0 to ADDR15 ...............
ADSR_0 and ADSR_1............

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

ICMR
ICR0
ICSR
ICSR1
ICSR2
ICSR3
IFCR
IMR
IPRA, IPRD to IPRF and IPRH t
IPRL
IRQCR
IRQSR
IRR
MBIMR0
MCR
MRA
MRB
NF2CYC
OCSR1
OCSR2
OSCCR
PACRL1
PACRL2
PACRL3
PACRL4
PADRL
PAIORL
PAPRL
PBCRL1

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$^2$C2) .......................................... 655
SCBRR (SCI) .......................................... 553
SCRDR .......................................... 537
SCRSR (SCI) .......................................... 537
SCSCR (SCI) .......................................... 541
SCSDCR .......................................... 552
SCSMR (SCI) .......................................... 538
SCSPTR (SCI) .......................................... 550

SSTDR0 to SSTDR3..........
SSTRSR ..........
STBCR1..........
STBCR2..........
STBCR3..........
STBCR4..........
STBCR5..........
STBCR6..........
TADCOBRA_4..........
TADCOBRB_4..........
TADCORA_4 ..........
TADCORB_4 ..........
TADCR ..........
TBTER..........
TBTM ..........
TCBR ..........
TCDR..........
TCNT ..........
TCNTCMPCLR ..........
TCNTS ..........
TCR..........
TCSYSTR ..........
TDDR..........
TDER ..........
TEC ..........
TGCR..........
TGR ..........
TICCR..........
TIER..........

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

Synchronous serial communication
unit (SSU) .....................................
System control instructions...........

T

Target pins and conditions for
high-impedance control .................
Test mode settings .........................
Time quanta is defined...................
Transfer clock ................................
Transfer information read skip
function ..........................................
Transfer information writeback ski
function ..........................................
Trap instructions ............................

U

User boot mode..............................
User break controller (UBC)..........
User break interrupt .......................
User break on instruction fetch cyc
User MAT......................................
User program mode .......................
Using interval timer mode .............
Using watchdog timer mode ..........

V

Vector numbers and vector table
address offsets...............................

# 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.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, J

## RENESAS SALES OFFICES

http://www.rer

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, Ma
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7137

Hardware I

Renesas Electronics Corporation

REJ