

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

# SH72546R

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer SuperH<sup>TM</sup> RISC engine Family

SH72546R R5F72546RKBGV

#### Notice

- 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.
- 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.
- 3. 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.
- 4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
- Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
  - "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 etc.
  - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.

- 6. 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.
- 7. 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 systems manufactured by you.
- 8. 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.
- 9. 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. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.
- 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
- 11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
- 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

## **General Precautions on Handling of Product**

#### 1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.

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

#### 2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.

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

#### 3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.

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

#### 4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.

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

#### 5. Reading from/Writing to Reserved Bit of Each Register

Note: Treat the reserved bit of register used in each module as follows except in cases where the specifications for values which are read from or written to the bit are provided in the description.

The bit is always read as 0. The write value should be 0 or one, which has been read immediately before writing.

Writing the value, which has been read immediately before writing has the advantage of preventing the bit from being affected on its extended function when the function is assigned.

# Configuration of This Manual

This manual comprises the following items:

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

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

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

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

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

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

#### 11. Index

### **Preface**

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

Target Users: This manual was written for users who will be using this LSI in the design of

application systems. Target users are expected to understand the fundamentals of

electrical circuits, logical circuits, and microcontrollers.

Objective: This manual was written to explain the hardware functions and electrical

characteristics of this LSI to the target users.

Refer to the SH-2A, SH2A-FPU Software Manual for a detailed description of the

instruction set.

Notes on reading this manual:

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

In order to understand the details of the CPU's functions

Read the SH-2A, SH2A-FPU Software Manual.

In order to understand the details of a register when its name is known Read the index that is the final part of the manual to find the page number of the entry on the

register. The addresses, bits, and initial values of the registers are summarized in section 31,

List of Registers.

#### Examples

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

#### (1) Registers

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

Example: CMCSR\_0

#### (2) Bits

When bit names are given in this manual, the higher-order bits are to the left and the lower-order bits are to the right.

Example: CKS1, CKS0

#### (3) Numbers

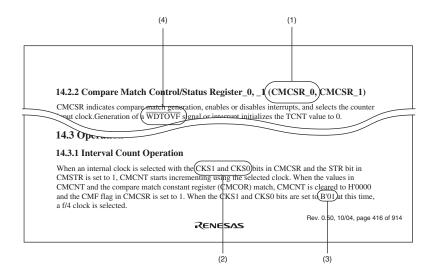
Binary numbers are given as B'xxxx, hexadecimal are given as H'xxxx, and decimal are given as xxxx.

Examples: B'11 or 11, H'EFA0, 1234

#### (4) Symbols

An overbar is added to the names of active-low signals.

Example: WDTOVF



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

Notation in bit figures and tables describing arrangements of bits

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

(1) Bit Indicates the bit number. In the case of a 32-bit register, the bits are arranged in order from 31 to 0, and in the case of a 16-bit register, the bits are arranged in the order from 15 to 0. (2) Bit Name The short form of the name of the bit or bit field within the register. When the individual bits of bit fields have to be clearly indicated, notation allowing this is included (e.g., ASID[3:0]). A reserved bit is indicated by -. Instead of a bit name, a blank is used for some bits, such as those of timer counters. Indicates the value of each bit after a power-on reset, i.e., the initial value. 0: Initial value is 0 1: Initial value is 1 -: Initial value is undefined Indicates whether each bit is readable or writable, or either writing to or reading from the bit is prohibited. The notation is as follows: R/W: Bit or field is readable and writable. R/(W): Bit or field is readable and writable. However, writing is only performed to clear the flag. Bit or field is readable and writable. However, "R" is indicated for all reserved bits. When writing to the bit is required, write the value stated in the bit table or the initial value. W: Bit or field is readable and writable. However, only the value in the bit table is guaranteed when reading from the bit. (5) Description Describes the function enabled by setting the bit. 15 10 2 0 13 11 IFE ASID Q ACMP[2:0] Initial value: Ω Ω Ω Ω Λ R/W R/W R/W R/W R/W R/W: R/W R R/W R/W R/W (1) (2)(3)(4)(5)Rit Bit Name Initial Value R/W Description 15 This bit is always read as 0. The write value should always be 0. 14 to 11 ASID ტიიი Address Identifier Enables or disables the pin function. 0 10 R Reserved This bit is always read as 0. This bit is always read as 1 Note: The bit names and sentences in the above figure are examples, and have nothing to do with the contents of this manual

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

# Contents

| Secti | on I     | Jverview                               | 1  |
|-------|----------|--|----|
| 1.1   | SH725    | 46R Features                           | 1  |
| 1.2   | Block I  | Diagram                                | 10 |
| 1.3   | Pin Des  | scriptions                             | 11 |
|       | 1.3.1    | Pin Arrangements                       | 11 |
|       | 1.3.2    | Pin Functions                          | 12 |
|       | 1.3.3    | List of Pins                           | 23 |
| Secti | on 2     | CPU                                    | 33 |
| 2.1   | Data Fo  | ormat                                  | 33 |
| 2.2   | Registe  | er Descriptions                        | 33 |
|       | 2.2.1    | General Registers                      | 33 |
|       | 2.2.2    | Control Registers                      | 34 |
|       | 2.2.3    | System Registers                       | 36 |
|       | 2.2.4    | Floating-Point Registers               | 37 |
|       | 2.2.5    | Floating-Point System Registers        | 38 |
|       | 2.2.6    | Register Bank                          | 41 |
|       | 2.2.7    | Initial Values of Registers            | 41 |
| 2.3   | Data Fo  | ormats                                 | 42 |
|       | 2.3.1    | Data Format in Registers               | 42 |
|       | 2.3.2    | Data Formats in Memory                 | 42 |
|       | 2.3.3    | Immediate Data Format                  | 43 |
| 2.4   | Instruct | tion Features                          | 44 |
|       | 2.4.1    | RISC-Type Instruction Set              | 44 |
|       | 2.4.2    | Addressing Modes                       | 48 |
|       | 2.4.3    | Instruction Format                     | 53 |
| 2.5   | Instruct | tion Set                               | 57 |
|       | 2.5.1    | Instruction Set by Classification      | 57 |
|       | 2.5.2    | Data Transfer Instructions             | 64 |
|       | 2.5.3    | Arithmetic Operation Instructions      | 68 |
|       | 2.5.4    | Logic Operation Instructions           | 71 |
|       | 2.5.5    | Shift Instructions                     | 72 |
|       | 2.5.6    | Branch Instructions                    | 73 |
|       | 2.5.7    | System Control Instructions            | 75 |
|       | 2.5.8    | Floating-Point Operation Instructions. | 77 |
|       | 2.5.9    | FPU-Related CPU Instructions           | 79 |
|       |          |  |    |

|      | 2.5.10 Bit Manipulation Instructions  | 79    |
|------|---|-------|
| 2.6  | Processing States.  | 81    |
| Sect | tion 3 Operating Modes  | 83    |
|      | Types of Operating Modes and Selection  |       |
| 3.1  | Types of Operating Modes and Selection  | 83    |
| Sect | tion 4 Clock Pulse Generator (CPG)  | 85    |
| 4.1  | Overview  | 85    |
| 4.2  | Pin Configuration   | 86    |
| 4.3  | Frequency Ranges and Clock Selection  | 87    |
| 4.4  | Clock Source  | 88    |
|      | 4.4.1 Connecting Crystal Resonator  | 88    |
|      | 4.4.2 External Clock Input  | 89    |
| 4.5  | Usage Notes   |       |
|      | 4.5.1 Note on Board Design.   | 90    |
|      | 4.5.2 Note on Connecting Power Supply for PLL Oscillator                            | 91    |
| Sect | tion 5 Address Space  | 93    |
| Sect | tion 6 Reset  | 97    |
| 6.1  | Reset Operation   |       |
|      | 6.1.1 Reset at Power-On   |       |
|      | 6.1.2 Reset during Operation  |       |
|      | 6.1.3 On-Chip RAM Data Retention during Reset                                       |       |
| 6.2  | Internal State after Reset Cancellation   |       |
| Sect | tion 7 Exception Handling   | 101   |
| 7.1  | Overview  |       |
| 7.1  | 7.1.1 Types of Exception Handling and Priority                                      |       |
|      | 7.1.1 Types of Exception Handling and Friority  7.1.2 Exception Handling Operations |       |
|      | 7.1.2 Exception Handling Operations  7.1.3 Exception Handling Vector Table          |       |
| 7.2  | Resets  |       |
| 1.2  | 7.2.1 Types of Reset  |       |
|      | 7.2.1 Types of Reset  |       |
| 7.3  | Address Errors  |       |
| 1.3  | 7.3.1 Address Error Sources   |       |
|      |   |       |
| 7.4  | 7.3.2 Address Error Exception Handling  |       |
| 7.4  | e   |       |
|      | 7.4.1 Register Bank Error Sources.  |       |
|      | 7.4.2 Register Bank Error Exception Handling  | 1 1 1 |

| 7.5  | Interru | ıpts  | 112 |
|------|---------|---|-----|
|      | 7.5.1   | Interrupt Sources   | 112 |
|      | 7.5.2   | Interrupt Priority Level  | 113 |
|      | 7.5.3   | Interrupt Exception Handling  | 114 |
| 7.6  | Excep   | tions Triggered by Instructions                                       | 115 |
|      | 7.6.1   | Types of Exceptions Triggered by Instructions                         | 115 |
|      | 7.6.2   | Trap Instructions   | 116 |
|      | 7.6.3   | Slot Illegal Exceptions   | 116 |
|      | 7.6.4   | General Illegal Instructions  | 117 |
|      | 7.6.5   | Integer Division Exceptions   | 117 |
|      | 7.6.6   | FPU Exceptions  |     |
| 7.7  | When    | Exception Sources Are Not Accepted                                    | 119 |
| 7.8  | Stack   | Status after Exception Handling Ends                                  | 120 |
| 7.9  | Usage   | Notes   | 122 |
|      | 7.9.1   | Value of Stack Pointer (SP)   | 122 |
|      | 7.9.2   | Value of Vector Base Register (VBR)                                   | 122 |
|      | 7.9.3   | Address Errors Caused by Stacking of Address Error Exception Handling | 122 |
| Sect | tion 8  | Interrupt Controller (INTC)   | 123 |
| 8.1  |         | res   |     |
| 8.2  | Input/  | Output Pins   | 125 |
| 8.3  | Regist  | er Descriptions   | 125 |
|      | 8.3.1   | Interrupt Priority Registers 01 to 29 (IPR01 to IPR29)                |     |
|      | 8.3.2   | Interrupt Control Register 0 (ICR0)                                   | 130 |
|      | 8.3.3   | Interrupt Control Register 1 (ICR1)                                   | 131 |
|      | 8.3.4   | IRQ Interrupt Request Register (IRQRR)                                | 132 |
|      | 8.3.5   | Bank Control Register (IBCR)  | 134 |
|      | 8.3.6   | Bank Number Register (IBNR)   | 135 |
|      | 8.3.7   | Software Interrupt Registers 1 to 15 (SINTR1 to SINTR15)              | 137 |
| 8.4  | Interru | ıpt Sources   | 138 |
|      | 8.4.1   | NMI Interrupt   | 138 |
|      | 8.4.2   | User Break Interrupt  | 138 |
|      | 8.4.3   | IRQ Interrupts  | 138 |
|      | 8.4.4   | Memory Error Interrupt  | 139 |
|      | 8.4.5   | Software Interrupts (SINT)  | 139 |
|      | 8.4.6   | On-Chip Peripheral Module Interrupts                                  | 139 |
| 8.5  |         | pt Exception Handling Vector Table and Priority                       |     |
| 8.6  | Opera   | tion  |     |
|      | 8.6.1   | Interrupt Operation Sequence  |     |
|      | 8.6.2   | Stack after Interrupt Exception Handling                              | 161 |

| 8.7  | Interru               | pt Response Time   | 162 |
|------|-----------------------|--|-----|
| 8.8  | Register Banks        |  |     |
|      | 8.8.1                 | Banked Registers and Input/Output Method                               | 168 |
|      | 8.8.2                 | Bank Saving and Restoring Operations                                   | 168 |
|      | 8.8.3                 | Saving and Restoring Operations after Saving Registers to All Banks    | 170 |
|      | 8.8.4                 | Register Bank Exception  | 171 |
|      | 8.8.5                 | Register Bank Error Exception Handling                                 | 171 |
| 8.9  | Data T                | ransfer with Interrupt Request Signals                                 | 172 |
|      | 8.9.1                 | Interrupt Request Signals as Sources for CPU (Not for Activating DMAC) | 173 |
|      | 8.9.2                 | Interrupt Request Signals as Sources for Activating DMAC (Not for CPU) | 173 |
|      | 8.9.3                 | Interrupt Request Signals as Sources for Activating A-DMAC             |     |
|      |                       | (Not for CPU)  | 174 |
| 8.10 | Usage                 | Note   | 174 |
|      | 8.10.1                | Timing to Clear Interrupt Source                                       | 174 |
| Sect | ion 9                 | User Break Controller (UBC)  | 175 |
| 9.1  |                       | es.  |     |
| 9.2  | Input/C               | Output Pin   | 177 |
| 9.3  | Register Descriptions |  |     |
|      | 9.3.1                 | Break Address Register_0 (BAR_0)                                       |     |
|      | 9.3.2                 | Break Address Mask Register_0 (BAMR_0)                                 |     |
|      | 9.3.3                 | Break Bus Cycle Register_0 (BBR_0)                                     |     |
|      | 9.3.4                 | Break Address Register_1 (BAR_1)                                       |     |
|      | 9.3.5                 | Break Address Mask Register_1 (BAMR_1)                                 |     |
|      | 9.3.6                 | Break Bus Cycle Register_1 (BBR_1)                                     |     |
|      | 9.3.7                 | Break Address Register_2 (BAR_2)                                       |     |
|      | 9.3.8                 | Break Address Mask Register_2 (BAMR_2)                                 |     |
|      | 9.3.9                 | Break Bus Cycle Register_2 (BBR_2)                                     |     |
|      | 9.3.10                | Break Address Register_3 (BAR_3)                                       |     |
|      | 9.3.11                | Break Address Mask Register_3 (BAMR_3)                                 |     |
|      | 9.3.12                | Break Bus Cycle Register_3 (BBR_3)                                     |     |
|      | 9.3.13                | Break Control Register (BRCR)  |     |
| 9.4  | Operat                | ion  | 200 |
|      | 9.4.1                 | Flow of the User Break Operation                                       |     |
|      | 9.4.2                 | Break on Instruction Fetch Cycle                                       |     |
|      | 9.4.3                 | Break on Data Access Cycle   |     |
|      | 9.4.4                 | Value of Saved Program Counter   |     |
|      | 9.4.5                 | Usage Examples   |     |
| 9.5  | Usage                 | Notes  | 207 |

| Sect | ion 10  | Bus State Controller (BSC)                                       | 209 |
|------|---------|--|-----|
| 10.1 | Feature | S  | 209 |
| 10.2 | Input/C | Output Pins  | 211 |
| 10.3 | Area O  | verview  | 212 |
|      | 10.3.1  | Address Map  | 212 |
|      | 10.3.2  | Operating Modes and Data Bus Width in CS0 Space                  | 214 |
| 10.4 | _       | er Descriptions  |     |
|      | 10.4.1  | CSn Space Bus Control Register (CSnBCR) (n = 0 to 3)             | 215 |
|      |         | CSn Space Wait Control Register (CSnWCR) (n = 0 to 3)            |     |
| 10.5 | Operati | ion  | 225 |
|      | 10.5.1  | Endian/Access Size and Data Alignment                            | 225 |
|      | 10.5.2  | External Space Interface   | 227 |
|      | 10.5.3  | Access Wait Control  | 233 |
|      | 10.5.4  | CSn Assert Period Expansion                                      | 235 |
|      | 10.5.5  | SRAM Interface with Byte Selection                               | 236 |
|      | 10.5.6  | Wait between Access Cycles                                       | 242 |
|      |         | Others   |     |
| 10.6 | SRAM    | Access Timing  | 246 |
|      | 10.6.1  | Standard SRAM  | 246 |
|      | 10.6.2  | SRAM with Byte Selection (BAS = 0)                               | 246 |
|      | 10.6.3  | SRAM with Byte Selection (BAS = 1)                               | 246 |
| Sect | ion 11  | Direct Memory Access Controller (DMAC)                           | 251 |
| 11.1 | Feature | S  | 251 |
| 11.2 | Registe | er Descriptions  | 254 |
|      | 11.2.1  | DMA Source Address Registers 0 to 7 (SAR0 to SAR7)               | 260 |
|      | 11.2.2  | DMA Destination Address Registers 0 to 7 (DAR0 to DAR7)          | 260 |
|      | 11.2.3  | DMA Transfer Count Registers 0 to 7 (DMATCR0 to DMATCR7)         | 261 |
|      | 11.2.4  | DMA Channel Control Registers 0 to 7 (CHCR0 to CHCR7)            | 261 |
|      | 11.2.5  | DMA Channel Flag Bit Registers 0 to 7 (CHFR0 to CHFR7)           | 267 |
|      |         | DMA TE Flag Mask Setting Registers 0 to 7 (TEMSK0 to TEMSK7)     |     |
|      | 11.2.7  | DMA Reload Source Address Registers 0 to 7 (RSAR0 to RSAR7)      | 272 |
|      | 11.2.8  | DMA Reload Destination Address Registers 0 to 7 (RDAR0 to RDAR7) | 273 |
|      | 11.2.9  | DMA Reload Transfer Count Registers 0 to 7                       |     |
|      |         | (RDMATCR0 to RDMATCR7)   | 274 |
|      | 11.2.10 | DMA Address Reload Count Registers 4 to 7 (ARCR4 to ARCR7)       | 275 |
|      | 11.2.11 | DMA Reload-Address Reload Count Registers 4 to 7                 |     |
|      |         | (RARCR4 to RARCR7)   | 276 |
|      | 11.2.12 | 2 DMA Operation Register (DMAOR)                                 | 277 |
|      | 11.2.13 | DMA Operation Flag Bit Register (DMAFR)                          | 280 |

|      | 11.2.14 | DMA Extension Resource Selectors 0 to 3 (DMARS0 to DMARS3) | 282 |
|------|---------|--|-----|
| 11.3 | Operat  | ion  | 285 |
|      | 11.3.1  | Transfer Flow  | 285 |
|      | 11.3.2  | DMA Transfer Requests                                      | 288 |
|      | 11.3.3  | Channel Priority   | 292 |
|      | 11.3.4  | DMA Transfer Types   | 296 |
| 11.4 | Special | l Operations   | 303 |
|      | 11.4.1  | Address Error Operation                                    | 303 |
|      | 11.4.2  | Operation on NMI   | 303 |
|      | 11.4.3  | Operation of Reloading Function 1                          | 303 |
|      | 11.4.4  | Operation of Reloading Function 2                          | 304 |
|      | 11.4.5  | Interface with On-Chip Peripheral Module                   | 304 |
|      | 11.4.6  | Integer to Floating-Point Conversion Operation             | 304 |
| 11.5 | Usage   | Note   | 307 |
| Sect | ion 12  | Automotive Direct Memory Access Controller (A-DMAC)        | 309 |
| 12.1 | Feature | 28   | 309 |
|      | 12.1.1  | Input/Output Pins  | 314 |
| 12.2 | Registe | er Descriptions  | 315 |
|      | 12.2.1  | A-DMAC Operation Register (ADMAOR)                         | 321 |
|      | 12.2.2  | A-DMAC Alias Base Register (ADMAABR)                       | 322 |
|      | 12.2.3  | A-DMAC Interrupt Control Registers (ADMAIE)                | 323 |
|      | 12.2.4  | A-DMAC Data Valid Registers (ADMADV)                       | 324 |
|      | 12.2.5  | A-DMAC Transfer End Registers (ADMATE)                     | 325 |
|      | 12.2.6  | A-DMAC Enable Registers (ADMADE)                           | 326 |
|      | 12.2.7  | A-DMAC Transfer Mode Registers (ADMAMODE)                  | 327 |
|      |         | A-DMAC Transfer Count Registers (ADMATCR)                  |     |
|      | 12.2.9  | A-DMAC Alias Pointer Registers (ADMAAR)                    | 329 |
|      | 12.2.10 | A-DMAC Reload Transfer Count Registers (ADMARTCR)          | 330 |
|      | 12.2.11 | A-DMAC Reload Alias Pointer Registers (ADMARAR)            | 330 |
|      | 12.2.12 | 2 A-DMAC Buffer Registers (ADMABUF)                        | 331 |
|      | 12.2.13 | 3 A-DMAC Receive Wait Registers (ADMARVPR)                 | 332 |
|      | 12.2.14 | A-DMAC Transmit Wait Registers (ADMATVPR)                  | 333 |
| 12.3 | Operat  | ion  | 334 |
|      | 12.3.1  | Alias Areas  | 334 |
|      | 12.3.2  | Each A-DMAC Channel Operation                              | 335 |
|      | 12.3.3  | Transfer Suspension and Resumption                         | 351 |
|      | 12.3.4  | Bus Operation in Data Transfer by A-DMAC                   | 352 |
|      | 12.3.5  | Channel Priorities   | 357 |
|      | 12.3.6  | Transfer Enable/Disable Conditions and Interrupt Requests  | 357 |

| 12.4  | Usage Note  | 360 |
|-------|---|-----|
| Secti | ion 13 Advanced Timer Unit III (ATU-III)                    | 361 |
| 13.1  | Features  |     |
| 13.2  | Register Addresses  | 365 |
| 13.3  | Input/Output Pins   | 386 |
| 13.4  | Overview of Common Controller                               | 387 |
|       | 13.4.1 Clock Bus.   | 387 |
| 13.5  | Register Description of Common Controller                   | 388 |
|       | 13.5.1 ATU-III Master Enable Register (ATUENR)              | 388 |
|       | 13.5.2 Clock Bus Control Register (CBCNT)                   | 391 |
|       | 13.5.3 Noise Cancellation Mode Register (NCMR)              | 393 |
| 13.6  | Overview of Prescalers                                      | 397 |
| 13.7  | Register Description of Prescalers                          | 398 |
|       | 13.7.1 Prescaler Registers 0 to 3 (PSCR0 to PSCR3)          | 398 |
| 13.8  | Operation of Prescalers                                     | 398 |
|       | 13.8.1 Starting Prescalers                                  | 398 |
|       | 13.8.2 Stopping and Restarting Operation                    | 399 |
| 13.9  | Overview of Timer A   | 400 |
|       | 13.9.1 Block Diagram of Timer A                             | 401 |
| 13.10 | Description of Timer A Registers                            |     |
|       | 13.10.1 Timer Control Register A (TCRA)                     | 402 |
|       | 13.10.2 Timer I/O Control Register 1A (TIOR1A)              |     |
|       | 13.10.3 Timer I/O Control Register 2A (TIOR2A)              |     |
|       | 13.10.4 Timer Status Register A (TSRA)                      | 407 |
|       | 13.10.5 Timer Interrupt Enable Register A (TIERA)           |     |
|       | 13.10.6 Input Capture Registers A0 to A5 (ICRA0 to ICRA5)   | 410 |
|       | 13.10.7 Free-Running Counter A (TCNTA)                      | 411 |
|       | 13.10.8 Noise Canceler Counters A0 to A5 (NCNTA0 to NCNTA5) |     |
|       | 13.10.9 Noise Canceler Registers A0 to A5 (NCRA0 to NCRA5)  | 414 |
| 13.11 | Operations of Timer A                                       | 415 |
|       | 13.11.1 Operation of Noise Canceler                         | 415 |
|       | 13.11.2 Operation of Free-Running Counter                   | 418 |
|       | 13.11.3 Input Capture                                       | 419 |
|       | 13.11.4 DMA Transfer  | 421 |
| 13.12 | Overview of Timer B   | 421 |
|       | 13.12.1 Block Diagram of Timer B                            | 422 |
| 13.13 | Descriptions of Timer B Registers                           | 424 |
|       | 13.13.1 Timer Control Register B (TCRB)                     | 424 |
|       | 13.13.2 Timer I/O Control Register B (TIORB)                | 425 |

|       | 13.13.3 Timer Status Register B (TSRB)                                    | 427 |
|-------|---|-----|
|       | 13.13.4 Timer Interrupt Enable Register B (TIERB)                         |     |
|       | 13.13.5 Edge Interval Measuring Counter B0 (TCNTB0)                       | 431 |
|       | 13.13.6 Input Capture Register B0 (ICRB0)                                 | 431 |
|       | 13.13.7 Output Compare Register B0 (OCRB0)                                | 432 |
|       | 13.13.8 Event Counter B1 (TCNTB1)   | 433 |
|       | 13.13.9 Output Compare Register B1 (OCRB1)                                | 433 |
|       | 13.13.10 Input Capture Register B1 (ICRB1)                                | 434 |
|       | 13.13.11 Input Capture Register B2 (ICRB2)                                | 435 |
|       | 13.13.12 Load Register B (LDB)  | 436 |
|       | 13.13.13 Reload Register B (RLDB)   | 437 |
|       | 13.13.14 Reloadable Counter B2 (TCNTB2)                                   | 438 |
|       | 13.13.15 Pulse Interval Multiplier Register (PIMR)                        | 439 |
|       | 13.13.16 Multiplied Clock Counter B6 (TCNTB6)                             | 440 |
|       | 13.13.17 Output Compare Register B6 (OCRB6)                               | 441 |
|       | 13.13.18 Output Compare Register B7 (OCRB7)                               |     |
|       | 13.13.19 Correcting Event Counter B3 (TCNTB3)                             | 443 |
|       | 13.13.20 Multiplied-and-Corrected Clock Counter B4 (TCNTB4)               |     |
|       | 13.13.21 Multiplied-and-Corrected Clock Generating Counter B5 (TCNTB5)    | 445 |
|       | 13.13.22 Correcting Counter Clearing Register B (TCCLRB)                  | 446 |
| 13.14 | Operations of Timer B   | 447 |
|       | 13.14.1 Edge Interval Measuring Function and Edge Input Stopping Function | 447 |
|       | 13.14.2 Frequency-Multiplied Clock Generator                              | 449 |
|       | 13.14.3 Frequency-Multiplied Clock Signal Corrector                       | 454 |
| 13.15 | Overview of Timer C   |     |
|       | 13.15.1 Block Diagram of Timer C  | 459 |
| 13.16 | Description of Timer C Registers  | 461 |
|       | 13.16.1 Timer Start Register C (TSTRC)                                    | 461 |
|       | 13.16.2 Noise Canceler Control Register C0 to C4 (NCCRC0 to NCCRC4)       | 462 |
|       | 13.16.3 Timer Control Registers C0 to C4 (TCRC0 to TCRC4)                 | 464 |
|       | 13.16.4 Timer Status Registers C0 to C4 (TSRC0 to TSRC4)                  |     |
|       | 13.16.5 Timer Interrupt Enable Registers C0 to C4 (TIERC0 to TIERC4)      | 470 |
|       | 13.16.6 Timer I/O Control Registers C0 to C4 (TIORC0 to TIORC4)           | 471 |
|       | 13.16.7 Timer Counters C0 to C4 (TCNTC0 to TCNTC4)                        | 474 |
|       | 13.16.8 General Registers C00 to C43 (GRC00 to GRC43)                     | 475 |
|       | 13.16.9 Noise Canceler Counters C00 to C43 (NCNTC00 to NCNTC43)           | 476 |
|       | 13.16.10 Noise Cancel Registers C00 to C43 (NCRC00 to NCRC43)             |     |
| 13.17 | Operations of Timer C   |     |
|       | 13.17.1 Input Capture Function  | 479 |
|       | 13.17.2 Compare Match Function  | 480 |

|       | 13.17.3 PWM Function   | 482 |
|-------|--|-----|
| 13.18 | Overview of Timer D  | 484 |
|       | 13.18.1 Block Diagram of Timer D                                     | 484 |
| 13.19 | Description of Timer D Registers                                     | 486 |
|       | 13.19.1 Timer Start Register (TSTRD)                                 |     |
|       | 13.19.2 Timer Control Registers D0 to D3 (TCRD0 to TCRD3)            | 487 |
|       | 13.19.3 Timer I/O Control Registers 1D0 to 1D3 (TIOR1D0 to TIOR1D3)  | 490 |
|       | 13.19.4 Timer I/O Control Registers 2D0 to 2D3 (TIOR2D0 to TIOR2D3)  | 493 |
|       | 13.19.5 Down Counter Starting Registers D0 to D3 (DSTRD0 to DSTRD3)  | 496 |
|       | 13.19.6 Down Counter Status Registers D0 to D3 (DSRD0 to DSRD3)      | 497 |
|       | 13.19.7 Down Counter Control Registers D0 to D3 (DCRD0 to DCRD3)     | 499 |
|       | 13.19.8 Timer Status Registers D0 to D3 (TSRD0 to TSRD3)             | 501 |
|       | 13.19.9 Timer Interrupt Enable Registers D0 to D3 (TIERD0 to TIERD3) | 507 |
|       | 13.19.10 Compare Match Pulse Output Control Registers D0 and D1      |     |
|       | (CMPOD0 and CMPOD1)  | 509 |
|       | 13.19.11 Timer Output Control Registers D0 to D3 (TOCRD0 to TOCRD3)  | 510 |
|       | 13.19.12 Timer Offset Base Registers D0 to D3 (OSBRD0 to OSBRD3)     | 511 |
|       | 13.19.13 Timer Counter 1D0 to 1D3 (TCNT1D0 to TCNT1D3)               | 512 |
|       | 13.19.14 Timer Counters 2D0 to 2D3 (TCNT2D0 to TCNT2D3)              | 513 |
|       | 13.19.15 Output Compare Registers D00 to D33 (OCRD00 to OCRD33)      | 514 |
|       | 13.19.16 General Registers D00 to D33 (GRD00 to GRD33)               | 515 |
|       | 13.19.17 Timer Down Counters D00 to 33 (DCNTD00 to DCNTD33)          | 516 |
| 13.20 | Operations of Timer D  | 518 |
| 13.21 | Overview of Timer E  |     |
|       | 13.21.1 Block Diagram of Timer E                                     |     |
| 13.22 | Description of Timer E Registers                                     |     |
|       | 13.22.1 Timer Start Register E (TSTRE)                               |     |
|       | 13.22.2 Subblock Starting Registers E0 to E5 (SSTRE0 to SSTRE5)      | 528 |
|       | 13.22.3 Prescaler Registers E0 to E5 (PSCRE0 to PSCRE5)              | 530 |
|       | 13.22.4 Timer Control Register E0 to E5 (TCRE0 to TCRE5)             | 531 |
|       | 13.22.5 Reload Control Registers E0 to E5 (RLDCRE0 to RLDCRE5)       | 532 |
|       | 13.22.6 Timer Status Registers E0 to E5 (TSRE0 to TSRE5)             | 532 |
|       | 13.22.7 Timer Interrupt Enable Registers E0 to E5 (TIERE0 to TIERE5) | 535 |
|       | 13.22.8 Timer Output Control Registers E0 to E5 (TOCRE0 to TOCRE5)   | 536 |
|       | 13.22.9 Timer Counters E00 to E53 (TCNTE00 to TCNTE53)               | 537 |
|       | 13.22.10 Cycle-Setting Registers E00 to E53 (CYLRE00 to CYLRE53)     | 538 |
|       | 13.22.11 Duty Cycle Setting Registers E00 to E53 (DTRE00 to DTRE53)  |     |
|       | 13.22.12 Cycle Reload Registers E00 to E53 (CRLDE00 to CRLDE53)      | 539 |
|       | 13.22.13 Duty Cycle Reload Registers E00 to E53 (DRLDE00 to DRLDE53) | 540 |
| 13.23 | Operations of Timer E  | 541 |

| 13.24 | Overview of Timer F  | . 545 |
|-------|--|-------|
|       | 13.24.1 Block Diagram  | . 546 |
|       | 13.24.2 Interrupts   | . 547 |
| 13.25 | Description of Timer F Registers                                       | . 548 |
|       | 13.25.1 Timer Start Register F (TSTRF)                                 | . 548 |
|       | 13.25.2 Noise Canceller Control Register F (NCCRF)                     | . 550 |
|       | 13.25.3 Timer Control Registers F0 to F19 (TCRF0 to TCRF19)            | . 552 |
|       | 13.25.4 Timer Interrupt Enable Registers F0 to F19 (TIERF0 to TIERF19) | . 555 |
|       | 13.25.5 Timer Status Registers F0 to F19 (TSRF0 to TSRF19)             | . 556 |
|       | 13.25.6 Timer Counters AF0 to AF19 (ECNTAF0 to ECNTAF19)               | . 559 |
|       | 13.25.7 Event Counters BF0 to BF19 (ECNTBF0 to ECNTBF19)               | . 560 |
|       | 13.25.8 Time Counters CF0 to CF19 (ECNTCF0 to ECNTCF19)                | . 562 |
|       | 13.25.9 General Registers AF0 to AF19 (GRAF0 to GRAF19)                | . 563 |
|       | 13.25.10 General Registers BF0 to BF19 (GRBF0 to GRBF19)               | . 564 |
|       | 13.25.11 General Registers CF0 to CF19 (GRCF0 to GRCF19)               | . 565 |
|       | 13.25.12 General Registers DF12 to DF15 (GRDF12 to GRDF15)             |       |
|       | 13.25.13 Capture Output Registers F0 to F19 (CDRF0 to CDRF19)          |       |
|       | 13.25.14 Noise Canceler Counters FA0 to FA19 (NCNTFA0 to NCNTFA19)     | . 568 |
|       | 13.25.15 Noise Canceler Counters FB0 to FB2 (NCNTFB0 to NCNTFB2)       | . 570 |
|       | 13.25.16 Noise Cancel Registers FA0 to FA19 (NCRFA0 to NCRFA19)        | . 572 |
|       | 13.25.17 Noise Cancel Registers FB0 to FB2 (NCRFB0 to NCRFB2)          | . 573 |
| 13.26 | Operations of Timer F  | . 574 |
|       | 13.26.1 Edge Counting  | . 574 |
|       | 13.26.2 Valid Edge Interval Counting                                   | . 576 |
|       | 13.26.3 Measurement of Time during High/Low Input Levels               | . 577 |
|       | 13.26.4 Measurement of PWM Input Waveform Timing                       | . 579 |
|       | 13.26.5 Rotation Speed/Pulse Measurement                               | . 581 |
|       | 13.26.6 Up/Down Event Count  | . 584 |
|       | 13.26.7 Four-time Multiplication Event Count                           | . 586 |
|       | 13.26.8 Overflow and Underflow   | . 588 |
| 13.27 | Overview of Timer G  | . 589 |
|       | 13.27.1 Block Diagram of Robots  | . 589 |
|       | 13.27.2 Interrupt Requests   | . 589 |
| 13.28 | Description of Timer G Registers                                       | . 590 |
|       | 13.28.1 Timer Start Register G (TSTRG)                                 | . 590 |
|       | 13.28.2 Timer Control Register G0 to G5 (TCRG0 to TCRG5)               | . 591 |
|       | 13.28.3 Timer Status Registers G0 to G5 (TSRG0 to TSRG5)               | . 592 |
|       | 13.28.4 Timer Counters G0 to G5 (TCNTG0 to TCNTG5)                     |       |
|       | 13.28.5 Compare Match Registers G0 to G5 (OCRG0 to OCRG5)              | . 595 |
| 13.29 | Operations of Timer G  | . 596 |
|       |  |       |

| 13.30 Overview of Timer H  | 597 |
|--|-----|
| 13.30.1 Block Diagram of Timer H                                       | 597 |
| 13.30.2 Interrupts   | 597 |
| 13.31 Description of Timer H Registers                                 | 598 |
| 13.31.1 Timer Control Register H (TCRH)                                | 598 |
| 13.31.2 Timer Status Register H (TSRH)                                 |     |
| 13.31.3 Timer Counter 1H (TCNT1H)                                      | 601 |
| 13.31.4 Compare Match Register 1H (OCR1H)                              |     |
| 13.31.5 Timer Counter 2H (TCNT2H)                                      | 603 |
| 13.32 Operations of Timer H  | 604 |
| 13.33 Overview of Timer J  | 605 |
| 13.33.1 Block Diagram of Timer J                                       | 605 |
| 13.34 Description of Timer J Registers                                 | 606 |
| 13.34.1 Timer Start Register J (TSTRJ)                                 | 606 |
| 13.34.2 Timer Control Registers J0 and J1 (TCRJ0 and TCRJ1)            | 607 |
| 13.34.3 FIFO Control Registers J0 and J1 (FCRJ0 and FCRJ1)             | 610 |
| 13.34.4 Timer Status Register J0 and J1 (TSRJ0 and TSRJ1)              | 612 |
| 13.34.5 Timer Interrupt Enable Registers J0 and J1 (TIERJ0 and TIERJ1) | 616 |
| 13.34.6 Timer Counter J0 and J1 (TCNTJ0 and TCNTJ1)                    | 617 |
| 13.34.7 Compare Match Registers J0 and J1 (OCRJ0 and OCRJ1)            | 617 |
| 13.34.8 FIFO Registers J0 and J1 (FIFOJ0 and FIFOJ1)                   | 618 |
| 13.34.9 FIFO Data Count Registers J0 and J1 (FDNRJ0 and FDNRJ1)        | 619 |
| 13.34.10 Noise Canceler Counters J0 and J1 (NCNTJ0 and NCNTJ1)         | 620 |
| 13.34.11 Noise Cancel Registers J0 and J1 (NCRJ0 and NCRJ1)            | 621 |
| 13.35 Operations of Timer J  | 622 |
| 13.36 Usage Notes  | 625 |
| 13.36.1 Input Capture Conflict Operation                               | 625 |
| 13.36.2 Compare Match Conflict Operation                               | 629 |
| 13.36.3 Load/Reload Conflict Operation                                 | 641 |
| 13.36.4 Counter Conflict Operation                                     | 644 |
| 13.36.5 Noise Canceler Conflict Operation                              | 649 |
| 13.36.6 Conflict Regarding Down Counter D                              | 652 |
| 13.36.7 Conflict between Timer B and Timer D                           | 657 |
| 13.36.8 Compare-Match Operation Specification                          | 661 |
| Section 14 Watchdog Timer (WDT)  |     |
| 14.1 Features  | 663 |
| 14.2 Input/Output Pin  | 665 |
| 14.3 Register Descriptions   | 665 |
| 14.3.1 Watchdog Timer Control Register (WTCR)                          | 665 |

|      | 14.3.2  | Watchdog Timer Counter (WTCNT)   | 668 |
|------|---------|--|-----|
|      | 14.3.3  | Watchdog Timer Status Register (WTSR)                                  | 669 |
|      | 14.3.4  | Watchdog Reset Control Register (WRCR)                                 | 671 |
|      | 14.3.5  | Notes on Register Access   | 673 |
| 14.4 | WDT U   | Jsage  | 674 |
|      | 14.4.1  | Using WDT in Watchdog Timer Mode                                       | 674 |
|      | 14.4.2  | Using WDT in Interval Timer Mode                                       | 676 |
| 14.5 | Usage   | Notes  | 677 |
|      | 14.5.1  | Timer Error  | 677 |
|      | 14.5.2  | Changing of Division Ratio   | 677 |
|      | 14.5.3  | Switching between Watchdog and Interval Timer Modes                    | 677 |
|      | 14.5.4  | System Reset by WDTOVF Signal  | 678 |
| Sect | ion 15  | Compare Match Timer (CMT)  | 679 |
| 15.1 | Feature | PS   | 679 |
| 15.2 | Registe | er Descriptions  | 680 |
|      | 15.2.1  | Compare Match Timer Start Register (CMSTR)                             | 681 |
|      | 15.2.2  | Compare Match Timer Control Register (CMCR)                            | 682 |
|      | 15.2.3  | Compare Match Timer Status Register (CMSR)                             | 683 |
|      | 15.2.4  | Compare Match Counter (CMCNT)  | 684 |
|      | 15.2.5  | Compare Match Constant Register (CMCOR)                                | 684 |
| 15.3 | Operat  | ion  | 685 |
|      | 15.3.1  | Interval Count Operation   | 685 |
|      | 15.3.2  | CMCNT Count Timing   | 685 |
| 15.4 | Interru | pts  | 686 |
|      | 15.4.1  | Interrupt Sources and DMA Transfer Requests                            | 686 |
|      | 15.4.2  | Timing of Compare Match Flag Setting                                   | 686 |
|      | 15.4.3  | Timing of Compare Match Flag Clearing                                  | 687 |
| 15.5 | Usage   | Notes  | 688 |
|      | 15.5.1  | Conflict between Write and Compare-Match Processes of CMCNT            | 688 |
|      | 15.5.2  | Conflict between Word-Write and Count-Up Processes of CMCNT            | 689 |
|      | 15.5.3  | Conflict between Setting of Compare-Match Flag and Clearing by the CPU | 690 |
| Sect | ion 16  | Serial Communications Interface (SCI)                                  | 691 |
| 16.1 |         | 28   |     |
| 16.2 |         | Output Pins  |     |
| 16.3 | _       | er Description   |     |
|      |         | Receive Shift Register (SCRSR1)  |     |
|      | 16.3.2  | Receive Data Register (SCRDR1)   | 696 |
|      | 1633    | Transmit Shift Register (SCTSR1)                                       | 696 |

|              | 16.3.4      | Transmit Data Register (SCTDR1)                                     | 697   |
|--------------|-------------|---|-------|
|              | 16.3.5      | Serial Mode Register (SCSMR1)                                       | 697   |
|              | 16.3.6      | Serial Control Register (SCSCR1)                                    | 700   |
|              | 16.3.7      | Serial Status Register (SCSSR1)                                     | 703   |
|              | 16.3.8      | Bit Rate Register (SCBRR1)  | 710   |
| 16.4         | Operat      | ion   | 716   |
|              | 16.4.1      | Overview  | 716   |
|              | 16.4.2      | Operation in Asynchronous Mode                                      | 718   |
|              | 16.4.3      | Clock Synchronous Mode  | 727   |
| 16.5         | SCI Int     | errupt Sources and A-DMAC   | 736   |
| 16.6         | Usage       | Notes   | 737   |
|              | 16.6.1      | SCTDR1 Writing and TDRE Flag  | 737   |
|              | 16.6.2      | Multiple Receive Error Occurrence                                   | 737   |
|              | 16.6.3      | Break Detection and Processing                                      | 738   |
|              | 16.6.4      | Sending a Break Signal  | 738   |
|              | 16.6.5      | Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) | 738   |
|              | 16.6.6      | Note on Using A-DMAC  | 740   |
|              | 16.6.7      | Note on Using External Clock in Clock Synchronous Mode              | 740   |
|              | 16.6.8      | Note on Using A-DMAC  | 740   |
|              |             | Serial Ports  |       |
|              | 16.6.10     | Note on Reception Only, with SCK Output, in Clock Synchronous Mode  | 741   |
| Soot         | ion 17      | Renesas Serial Peripheral Interface (RSPI)                          | 7/2   |
|              |             | es  |       |
| 17.1         |             |   |       |
| 17.2<br>17.3 | -           | Output Pinser Descriptions  |       |
| 17.3         | _           | RSPI Control Register (SPCR)  |       |
|              |             | RSPI Slave Select Polarity Register (SSLP)                          |       |
|              |             | RSPI Pin Control Register (SPPCR)                                   |       |
|              |             | RSPI Status Register (SPSR)   |       |
|              |             | RSPI Data Register (SPDR)   |       |
|              |             | RSPI Sequence Control Register (SPSCR)                              |       |
|              |             | RSPI Sequence Status Register (SPSSR)                               |       |
|              |             | RSPI Bit Rate Register (SPBR)                                       |       |
|              |             | RSPI Clock Delay Register (SPCKD)                                   |       |
|              |             | SPI Slave Select Negation Delay Register (SSLND)                    |       |
|              |             | RSPI Next-Access Delay Register (SPND)                              |       |
|              |             | 2 RSPI Command Register (SPCMD)                                     |       |
| 17.4         |             | ion   |       |
| ± / .⊤       | -           | Overview of RSPI Operations   |       |
|              | 1 / 1 1 1 1 |   | 1 1 0 |

|                                   | 17.4.2   | Controlling RSPI Pins  | 775 |
|-----------------------------------|----------|--|-----|
|                                   | 17.4.3   | RSPI System Configuration Example                                  | 776 |
|                                   | 17.4.4   | Transfer Format  | 783 |
|                                   | 17.4.5   | Data Format  | 785 |
|                                   | 17.4.6   | Transmission Buffer Empty/Receive Buffer Full Flags                | 790 |
|                                   | 17.4.7   | Error Detection  | 792 |
|                                   | 17.4.8   | Initializing RSPI  | 797 |
|                                   | 17.4.9   | Master Mode Operation  | 798 |
|                                   | 17.4.10  | Slave Mode Operation   | 806 |
|                                   | 17.4.11  | Loopback Mode  | 811 |
| Secti                             | on 18    | Controller Area Network (RCAN-TL1)                                 | 813 |
| 18.1                              | Summa    | ıry  | 813 |
|                                   | 18.1.1   | Overview   | 813 |
|                                   | 18.1.2   | Scope  | 813 |
|                                   | 18.1.3   | Audience   | 813 |
|                                   | 18.1.4   | References   | 813 |
|                                   | 18.1.5   | Features   | 814 |
| 18.2                              | Archite  | cture  | 815 |
| 18.3 Programming Model - Overview |          | nming Model - Overview   | 819 |
|                                   | 18.3.1   | Memory Map   | 819 |
|                                   | 18.3.2   | Mailbox Structure  | 821 |
|                                   | 18.3.3   | RCAN-TL1 Control Registers   | 839 |
|                                   | 18.3.4   | RCAN-TL1 Mailbox Registers   | 862 |
|                                   | 18.3.5   | Timer Registers  | 877 |
| 18.4                              | Applica  | ation Note   | 891 |
|                                   | 18.4.1   | Test Mode Settings   | 891 |
|                                   | 18.4.2   | Configuration of RCAN-TL1  | 893 |
|                                   | 18.4.3   | Message Transmission Sequence                                      | 897 |
|                                   | 18.4.4   | Message Receive Sequence   | 912 |
|                                   | 18.4.5   | Reconfiguration of Mailbox   | 914 |
| 18.5                              | Parity I | Detection  | 916 |
| 18.6                              |          | ot Sources   |     |
| 18.7                              | DMAC     | Interface  | 918 |
| 18.8                              | CAN B    | us Interface   | 919 |
| 18.9                              | A-DM     | AC Interface   | 919 |
| 18.10                             | Setting  | I/O Ports for RCAN-TL1   | 920 |
|                                   | _        | Notes  |     |
|                                   | _        | Notes on Port Setting for Multiple Channels Used as Single Channel |     |

| Sect | ion 19   | A/D Converter (ADC)   | 925 |
|------|----------|---|-----|
| 19.1 | Feature  | 98  | 925 |
| 19.2 | Input/C  | Output Pins   | 928 |
| 19.3 | Function | ons Assigned to Each Channel  | 931 |
| 19.4 | Registe  | er Descriptions   | 932 |
|      | 19.4.1   | A/D Data Registers 0 to 27, 40 to 48, Diag0, and Diag1                |     |
|      |          | (ADR0 to ADR27, ADR40 to ADR48, ADRD0, and ADRD1)                     | 935 |
|      | 19.4.2   | A/D Control Registers 0 and 1 (ADCSR0 and ADCSR1)                     | 940 |
|      | 19.4.3   | A/D Control Extended Registers 0 and 1 (ADCER0 and ADCER1)            | 943 |
|      | 19.4.4   | A/D Channel Select Registers 0, 1, and 3                              |     |
|      |          | (ADANS0, ADANS1, and ADANS3)  | 947 |
|      | 19.4.5   | A/D Conversion Status Registers 0 and 1 (ADREF0 and ADREF1)           | 949 |
|      | 19.4.6   | A/D-Converted Value Addition Mode Select Registers 0 and 1            |     |
|      |          | (ADADS0 and ADADS1)   | 951 |
|      | 19.4.7   | A/D-Converted Value Addition Count Select Registers 0 and 1           |     |
|      |          | (ADADC0 and ADADC1)   | 954 |
|      | 19.4.8   | A/D Interrupt Trigger Enable Registers 0 and 1 (ADTRE0 and ADTRE1)    | 955 |
|      | 19.4.9   | A/D Interrupt Trigger Source Select Registers 0 and 1                 |     |
|      |          | (ADTRS0 and ADTRS1)   | 957 |
|      | 19.4.10  | A/D Interrupt Software Trigger Registers 0 and 1                      |     |
|      |          | (ADSTRG0 and ADSTRG1)   | 960 |
|      | 19.4.11  | A/D Interrupt Trigger Conversion End Flag Registers 0 and 1           |     |
|      |          | (ADTRF0 and ADTRF1)   | 962 |
|      | 19.4.12  | 2 A/D Interrupt Trigger Conversion Interrupt Enable Registers 0 and 1 |     |
|      |          | (ADTRD0 and ADTRD1)   |     |
|      |          | 3 Interface with CPU  |     |
| 19.5 |          | ion   |     |
|      |          | Scan Conversion   |     |
|      |          | Single-Cycle Scan Conversion Mode                                     |     |
|      |          | Continuous Scan Conversion Mode                                       |     |
|      |          | Interrupt Conversion  |     |
|      |          | Example Operation of Interrupt Conversion                             |     |
|      |          | Interrupt Conversion during Scan Conversion                           |     |
|      |          | Analog Input Sampling and Scan Conversion Time                        |     |
|      |          | Starting Scan Conversion with External Trigger                        |     |
|      |          | Starting Scan Conversion with ATU-III Timer Trigger                   |     |
|      |          | Monitoring via ADEND_A and ADEND_B Output Pins                        |     |
| 19.6 |          | pt Sources and DMA Transfer Request                                   |     |
|      |          | Interrupt Requests on Completion of Scan Conversion                   |     |
|      | 19.6.2   | Interrupt Requests on Completion of Interrupt Conversion              | 983 |

| 19.7 | Definit        | ion of A/D Conversion Accuracy   | 984   |
|------|----------------|--|-------|
| 19.8 | Usage          | Notes  | 986   |
|      | 19.8.1         | Analog Input Voltage Range   | 986   |
|      | 19.8.2         | Relationship among AV <sub>cc</sub> , AV <sub>ss</sub> , V <sub>cc</sub> , and V <sub>ss</sub> | 986   |
|      | 19.8.3         | Allowable Settings for Pins AVrefh_A/AVrefh_B and AVrefl_A/AVrefl_B                            | 986   |
|      | 19.8.4         | Precautions on Board Design  | 987   |
| Sect | ion 20         | JTAG Interface   | 991   |
| 20.1 | Feature        | S  | 991   |
| 20.2 | Input/C        | Output Pins  | 993   |
| 20.3 | Registe        | er Descriptions  | 994   |
|      | 20.3.1         | Instruction Register (SDIR)  | 995   |
|      | 20.3.2         | ID Register (SDID)   | 995   |
|      | 20.3.3         | Bypass Register (SDBPR)  | 995   |
|      | 20.3.4         | Boundary Scan Register (SDBSR)   | 996   |
| 20.4 | Operat         | ions   | 1012  |
|      | 20.4.1         | TAP Controller   | 1012  |
|      | 20.4.2         | Supported Commands   | 1013  |
|      | 20.4.3         | Notes  | 1014  |
| 20.5 | Usage          | Notes  | 1016  |
| Sect | ion 21         | Advanced User Debugger II (AUD-II)   | .1017 |
| 21.1 | Feature        | PS   | 1017  |
| 21.2 | Input/C        | Output Pins  | 1020  |
|      | 21.2.1         | Commonly Used Pins   | 1020  |
|      | 21.2.2         | Pin Descriptions in AUD Tracing Mode   | 1021  |
|      | 21.2.3         | Pin Descriptions in RAM Monitoring Mode  | 1022  |
| 21.3 |                | racing Mode  |       |
|      | 21.3.1         | Register Descriptions  | 1023  |
|      | 21.3.2         | AUD Control Register (AUCSR)   | 1023  |
|      | 21.3.3         | AUD Window A Start Address Register (AUWASR)   | 1027  |
|      | 21.3.4         | AUD Window A End Address Register (AUWAER)   | 1027  |
|      |                | AUD Window B Start Address Register (AUWBSR)   |       |
|      |                | AUD Window B End Address Register (AUWBER)   |       |
|      | 21.3.7         | AUD Extended Control Register (AUECSR)   |       |
|      | 21.3.8         | Operation  |       |
|      |                | Usage Notes on AUD Tracing Mode  |       |
| 21.4 |                | Monitoring Mode  |       |
|      |                | Communication Protocol   |       |
|      |                | Operations   |       |
|      | · · · <b>-</b> | T  |       |

|      | 21.4.3  | Notes on RAM Monitoring Mode                         | 1049 |
|------|---------|--|------|
| Sect | ion 22  | Pin Function Controller (PFC)                        | 1051 |
| 22.1 | Registe | er Descriptions                                      | 1060 |
|      | 22.1.1  | Port A I/O Register (PAIOR)                          | 1062 |
|      | 22.1.2  | Port A Control Registers 1 to 4 (PACR1 to PACR4)     | 1063 |
|      | 22.1.3  | Port B I/O Register (PBIOR)                          | 1076 |
|      |         | Port B Control Registers 1 to 4 (PBCR1 to PBCR4)     |      |
|      | 22.1.5  | Port C I/O Register (PCIOR)                          | 1091 |
|      | 22.1.6  | Port C Control Registers 1 to 4 (PCCR1 to PCCR4)     | 1092 |
|      | 22.1.7  | Port D I/O Register (PDIOR)                          | 1107 |
|      |         | Port D Control Registers 1 and 2 (PDCR1 and PDCR2)   |      |
|      | 22.1.9  | Port E I/O Register (PEIOR)                          | 1112 |
|      | 22.1.10 | Port E Control Registers 1 and 2 (PECR1 and PECR2)   | 1113 |
|      | 22.1.11 | Port F I/O Register (PFIOR)                          | 1117 |
|      | 22.1.12 | 2 Port F Control Registers 1 and 2 (PFCR1 and PFCR2) | 1118 |
|      | 22.1.13 | 3 Port G I/O Register (PGIOR)                        | 1122 |
|      | 22.1.14 | Port G Control Registers 1 and 2 (PGCR1 and PGCR2)   | 1123 |
|      | 22.1.15 | 5 Port H I/O Register (PHIOR)                        | 1128 |
|      | 22.1.16 | Port H Control Register (PHCR)                       | 1129 |
|      | 22.1.17 | Port J I/O Register (PJIOR)                          | 1131 |
|      | 22.1.18 | B Port J Control Registers 1 and 2 (PJCR1 and PJCR2) | 1132 |
|      | 22.1.19 | Port K I/O Register (PKIOR)                          | 1135 |
|      | 22.1.20 | Port K Control Registers 1 and 2 (PKCR1 and PKCR2)   | 1136 |
|      | 22.1.21 | Port L I/O Register (PLIOR)                          | 1140 |
|      | 22.1.22 | 2 Port L Control Registers 1 and 2 (PLCR1 and PLCR2) | 1141 |
| Sect | ion 23  | I/O Ports  | 1145 |
| 23.1 | Overvi  | ew   | 1145 |
| 23.2 | Registe | er Descriptions                                      | 1146 |
| 23.3 | _       | <u>.</u>   |      |
|      | 23.3.1  | Port A Data Register (PADR)                          | 1148 |
|      |         | Port A Port Register (PAPR)                          |      |
| 23.4 |         |  |      |
|      | 23.4.1  | Port B Data Register (PBDR)                          | 1152 |
|      | 23.4.2  | Port B Port Register (PBPR)                          | 1154 |
|      |         | Port B Inverting Register (PBIR)                     |      |
|      |         | Port B Driving Ability Setting Register (PBDSR)      |      |
|      | 23.4.5  | Port B Pin State Setting Register (PBPSR)            | 1158 |
| 23.5 | Port C  |  |      |

|       | 23.5.1  | Port C Data Register (PCDR)                     | 1159 |
|-------|---------|---|------|
|       | 23.5.2  | Port C Port Register (PCPR)                     | 1161 |
| 23.6  | Port D. |   | 1163 |
|       | 23.6.1  | Port D Data Register (PDDR)                     | 1163 |
|       | 23.6.2  | Port D Port Register (PDPR)                     | 1165 |
|       | 23.6.3  | Port D Inverting Register (PDIR)                | 1166 |
| 23.7  | Port E. |   | 1167 |
|       | 23.7.1  | Port E Data Register (PEDR)                     | 1167 |
|       |         | Port E Port Register (PEPR)                     |      |
|       |         | Port E Inverting Register (PEIR)                |      |
|       | 23.7.4  | Port E Driving Ability Setting Register (PEDSR) | 1171 |
| 23.8  |         |   |      |
|       |         | Port F Data Register (PFDR)                     |      |
|       |         | Port F Port Register (PFPR)                     |      |
|       |         | Port F Inverting Register (PFIR)                |      |
|       |         | Port F Driving Ability Setting Register (PFDSR) |      |
|       | 23.8.5  | Port F Pin State Setting Register (PFPSR)       | 1178 |
| 23.9  | Port G. |   | 1179 |
|       |         | Port G Data Register (PGDR)                     |      |
|       |         | Port G Port Register (PGPR)                     |      |
|       |         | Port G Inverting Register (PGIR)                |      |
|       |         | Port G Driving Ability Setting Register (PGDSR) |      |
|       | 23.9.5  | Port G Edge Selecting Register (PGER)           | 1185 |
| 23.10 |         |   |      |
|       |         | Port H Data Register (PHDR)                     |      |
|       | 23.10.2 | Port H Port Register (PHPR)                     | 1189 |
| 23.11 |         |   |      |
|       | 23.11.1 | Port J Data Register (PJDR)                     | 1190 |
|       |         | Port J Port Register (PJPR)                     |      |
|       |         | Port J Inverting Register (PJIR)                |      |
|       |         | Port J Driving Ability Setting Register (PJDSR) |      |
|       | 23.11.5 | Port J Pin State Setting Register (PJPSR)       | 1196 |
| 23.12 | Port K. |   | 1198 |
|       | 23.12.1 | Port K Data Register (PKDR)                     | 1198 |
|       | 23.12.2 | Port K Port Register (PKPR)                     | 1200 |
|       | 23.12.3 | Port K Inverting Register (PKIR)                | 1201 |
|       | 23.12.4 | Port K Driving Ability Setting Register (PKDSR) | 1202 |
|       | 23.12.5 | Port K Pin State Setting Register (PKPSR)       | 1203 |
| 23.13 | Port L. |   | 1205 |
|       | 23.13.1 | Port L Data Register (PLDR)                     | 1205 |

|       |         | Port L Port Register (PLPR)   |      |
|-------|---------|---|------|
|       | 23.13.3 | Port L Inverting Register (PLIR)  | 1208 |
| 23.14 | CK Co   | ntrol Register (CKCR)   | 1209 |
| 23.15 | Port O  | atput Disable (POD)   | 1210 |
| 23.16 | Usage   | Note  | 1211 |
|       | 23.16.1 | Note on State Immediately after Reset                                   | 1211 |
|       | 23.16.2 | Note on Operation of Input/Output Pins on a Reset by an Internal Source | 1211 |
| Sect  | ion 24  | Multi-Input Signature Generator (MISG)                                  | 1213 |
| 24.1  | Overvi  | ew  | 1213 |
| 24.2  | _       | er Descriptions   |      |
|       | 24.2.1  | Calculation Data Register (MISRCDR)                                     | 1214 |
|       | 24.2.2  | Multi-Input Signature Register (MISR)                                   | 1215 |
|       | 24.2.3  | MISR Control Register (MISRCR)  | 1216 |
| Sect  | ion 25  | ROM   | 1217 |
| 25.1  | Feature | S.  | 1217 |
| 25.2  | Input/C | Output Pins   | 1221 |
| 25.3  | Registe | er Descriptions   | 1222 |
|       | 25.3.1  | Flash Pin Monitor Register (FPMON)                                      | 1223 |
|       | 25.3.2  | Flash Mode Register (FMODR)   | 1224 |
|       | 25.3.3  | Flash Access Status Register (FASTAT)                                   | 1225 |
|       | 25.3.4  | Flash Access Error Interrupt Enable Register (FAEINT)                   | 1228 |
|       | 25.3.5  | ROM MAT Select Register (ROMMAT)  | 1230 |
|       | 25.3.6  | FCU RAM Enable Register (FCURAME)                                       | 1231 |
|       | 25.3.7  | Flash Status Register 0 (FSTATR0)                                       | 1232 |
|       | 25.3.8  | Flash Status Register 1 (FSTATR1)                                       | 1236 |
|       | 25.3.9  | FCU RAM ECC Error Control Register (FRAMECCR)                           | 1238 |
|       | 25.3.10 | Flash P/E Mode Entry Register (FENTRYR)                                 | 1239 |
|       | 25.3.11 | Flash Protect Register (FPROTR)   | 1246 |
|       | 25.3.12 | Flash Reset Register (FRESETR)  | 1247 |
|       | 25.3.13 | FCU Command Register (FCMDR)  | 1248 |
|       | 25.3.14 | FCU Processing Switch Register (FCPSR)                                  | 1249 |
|       | 25.3.15 | Flash P/E Status Register (FPESTAT)                                     | 1250 |
| 25.4  | Overvi  | ew of ROM-Related Modes   | 1251 |
| 25.5  | Boot M  | Iode  | 1253 |
|       | 25.5.1  | System Configuration  | 1253 |
|       | 25.5.2  | State Transition in Boot Mode   | 1254 |
|       | 25.5.3  | Automatic Adjustment of Bit Rate  | 1256 |
|       | 25.5.4  | Inquiry/Selection Host Command Wait State                               | 1257 |

|      | 25.5.5  | Programming/Erasing Host Command Wait State                  | 1276 |
|------|---------|--|------|
| 25.6 | User P  | rogram Mode  | 1288 |
|      | 25.6.1  | FCU Command List   | 1288 |
|      | 25.6.2  | Conditions for FCU Command Acceptance                        | 1291 |
|      | 25.6.3  | FCU Command Usage  | 1296 |
|      | 25.6.4  | Suspending Operation   | 1312 |
| 25.7 | User B  | oot Mode   | 1315 |
|      | 25.7.1  | User Boot Mode Initiation                                    | 1315 |
|      | 25.7.2  | User MAT Programming   | 1317 |
| 25.8 | Protect | ion  | 1318 |
|      | 25.8.1  | Hardware Protection  | 1318 |
|      | 25.8.2  | Software Protection  | 1319 |
|      | 25.8.3  | Error Protection   | 1320 |
| 25.9 | Usage   | Notes  | 1323 |
|      | 25.9.1  | Switching between User MAT and User Boot MAT                 | 1323 |
|      | 25.9.2  | Other Notes  |      |
| Sact | ion 26  | EEPROM   | 1320 |
| 26.1 |         | ELI KOW  |      |
| 26.2 |         | Output Pins  |      |
| 26.3 | -       | er Descriptions  |      |
| 20.3 | _       | Flash Mode Register (FMODR)                                  |      |
|      |         | Flash Access Status Register (FASTAT)                        |      |
|      |         | Flash Access Error Interrupt Enable Register (FAEINT)        |      |
|      |         | EEPROM Read Enable Register 0 (EEPRE0)                       |      |
|      |         | EEPROM Read Enable Register 1 (EEPRE1)                       |      |
|      |         | EEPROM Program/Erase Enable Register 0 (EEPWE0)              |      |
|      |         | EEPROM Program/Erase Enable Register 1 (EEPWE1)              |      |
|      |         | Flash P/E Mode Entry Register (FENTRYR)                      |      |
|      |         | EEPROM Blank Check Control Register (EEPBCCNT)               |      |
|      |         | EEPROM Blank Check Control Register (EEPBCSTAT)              |      |
|      |         | EEPROM MAT Select Register (EEPMAT)                          |      |
| 26.4 |         | ew of EEPROM-Related Modes                                   |      |
| 26.5 |         |  |      |
| 20.3 |         | NodeSelection Host Commands                                  |      |
|      |         |  |      |
| 26.6 |         | Programming/Erasing Host Commands                            |      |
| 26.6 |         | Iode, User Program Mode, and User Boot Mode FCU Command List |      |
|      |         |  |      |
|      |         | Conditions for FCU Command Acceptance                        |      |
|      | ∠n.n. 3 | FUU Command Usage  | 1364 |

| 26.7 | Protect  | ion  | 1368 |
|------|----------|--|------|
|      | 26.7.1   | Hardware Protection                                      | 1368 |
|      | 26.7.2   | Software Protection                                      | 1368 |
|      | 26.7.3   | Error Protection   | 1369 |
| 26.8 | Produc   | t Information MAT  | 1371 |
| 26.9 | Usage    | Notes  | 1372 |
| Sect | ion 27   | ROM Cache (ROMC)   | 1375 |
| 27.1 | Feature  | es   | 1375 |
| 27.2 | Config   | uration  | 1375 |
| 27.3 | Registe  | er Descriptions  | 1378 |
|      | 27.3.1   | ROM Cache Control Register (RCCR)                        | 1378 |
|      | 27.3.2   | ROM Cache Control Register 2 (RCCR2)                     | 1380 |
| 27.4 | Operat   | ion  | 1384 |
|      | 27.4.1   | Data Cache Lookup  | 1384 |
|      | 27.4.2   | Instruction Cache Lookup                                 | 1385 |
|      | 27.4.3   | Hardware Prefetching                                     | 1386 |
| 27.5 | Usage    | Note   | 1388 |
| Sect | ion 28   | RAM  | 1389 |
| 28.1 | Feature  | es   | 1389 |
| 28.2 | Registe  | er Descriptions  | 1392 |
|      | 28.2.1   | RAM Enable Control Registers 0 and 1 (RAMEN0 and RAMEN1) | 1393 |
|      | 28.2.2   | RAM Write Enable Control Registers 0 and 1               |      |
|      |          | (RAMWEN0 and RAMWEN1)                                    | 1404 |
|      | 28.2.3   | RAM ECC Enable Control Register (RAMECC)                 | 1414 |
|      | 28.2.4   | RAM Error Status Register (RAMERR)                       | 1415 |
|      | 28.2.5   | RAM Error Interrupt Control Register (RAMINT)            | 1419 |
|      | 28.2.6   | RAM Access Cycle Set Register (RAMACYC)                  | 1422 |
|      | 28.2.7   | Notes on Register Access                                 | 1424 |
| 28.3 | On-Ch    | ip RAM Operations  | 1426 |
| 28.4 | RAM I    | Data Retention   | 1427 |
|      | 28.4.1   | Data Retention at Reset                                  | 1427 |
|      | 28.4.2   | Data Retention at Hardware Standby                       | 1427 |
| 28.5 | Notes of | on Usage   | 1429 |
|      | 28.5.1   | Page Conflict  | 1429 |
|      | 28.5.2   | State After Turning on Power                             | 1429 |
|      | 28.5.3   | Write Operation When Writing RAM is Disabled             | 1429 |

| Sect | ion 29             | Power-Down Modes  | 1431 |  |  |
|------|--------------------|---|------|--|--|
| 29.1 | Feature            | 28  | 1431 |  |  |
|      | 29.1.1             | Power-Down Modes  | 1431 |  |  |
| 29.2 | Input/C            | Output Pins   | 1433 |  |  |
| 29.3 | Registe            | er Descriptions   | 1433 |  |  |
|      | 29.3.1             | Standby Control Register (STBCR)                                    | 1433 |  |  |
|      | 29.3.2             | Note on Accessing STBCR   | 1436 |  |  |
| 29.4 | Operat             | ion   | 1437 |  |  |
|      | 29.4.1             | Hardware Standby Mode   | 1437 |  |  |
|      | 29.4.2             | Sleep Mode  | 1438 |  |  |
|      | 29.4.3             | Module Standby Function   | 1439 |  |  |
| Sect | ion 30             | Reliability   | 1441 |  |  |
| 30.1 | Reliabi            | lity  | 1441 |  |  |
| Sect | ion 31             | List of Registers   | 1443 |  |  |
| 31.1 |                    | er Addresses  |      |  |  |
|      | _                  | ed by module name, in ordered of the corresponding section numbers) | 1444 |  |  |
| 31.2 |                    | Register Bits   |      |  |  |
| 31.3 |                    | er States in Each Operating Mode                                    |      |  |  |
| Sect | ion 32             | Electrical Characteristics  | 1709 |  |  |
| 32.1 |                    | te Maximum Ratings  |      |  |  |
| 32.2 | DC Characteristics |   |      |  |  |
| 32.3 |                    | AC Characteristics  |      |  |  |
|      | 32.3.1             | Timing for Power On and Off   |      |  |  |
|      | 32.3.2             | Timing for Operation Mode and Oscillation                           |      |  |  |
|      | 32.3.3             | Clock Timing  |      |  |  |
|      |                    | Control Signal Timing   |      |  |  |
|      |                    | Bus Timing  |      |  |  |
|      |                    | Advanced Timer Unit Timing and Advanced Pulse Controller Timing     |      |  |  |
|      | 32.3.7             | I/O Port Timing.  | 1749 |  |  |
|      | 32.3.8             | Watchdog Timer Timing   | 1750 |  |  |
|      |                    | Serial Communications Interface Timing                              |      |  |  |
|      |                    | CAN Timing  |      |  |  |
|      |                    | SPI Timing  |      |  |  |
|      |                    | 2 A/D Converter Timing  |      |  |  |
|      |                    | B UBC Trigger Timing  |      |  |  |
|      | 32.3.14            | Output Slew Rate  | 1764 |  |  |

|      | 32.3.15 JTAG Interface Timing   | 1765 |
|------|---|------|
|      | 32.3.16 AUD Timing  | 1767 |
|      | 32.3.17 Measuring Conditions for AC Characteristics                     |      |
| 32.4 | A/D Converter Characteristics   | 1771 |
| 32.5 | Flash Memory Characteristics  | 1772 |
| 32.6 | EEPROM Characteristics  | 1774 |
| 32.7 | Usage Note  | 1775 |
|      | 32.7.1 Notes on Connecting External Capacitor for Current Stabilization | 1775 |
| App  | endix   | 1777 |
|      | Pin States  |      |
| B.   | Product Code Lineup   | 1784 |
| C.   | Package Dimensions  |      |
| Mai  | n Revisions and Additions in this Edition                               | 1787 |
| Inde | ex  | 1012 |

SH72546R Section 1 Overview

# Section 1 Overview

#### 1.1 SH72546R Features

The SH72546R is a single-chip microcontroller featuring an original Renesas RISC (reduced instruction set computer) SH-2A-type CPU core along with integrated peripheral functions essential to system configuration.

This CPU brings the user the ability to set up high-performance systems with strong functionality at less expense than was achievable with previous microcontrollers, and is even able to handle real-time control applications requiring high-speed characteristics. Superscalar architecture and a Harvard cache architecture further improve the CPU's characteristics and performance.

The SH-2A has a 32-bit RISC architecture, and features upward compatibility with the SH-2E core at the object code level. Furthermore, the new instructions added to the instruction set for the existing SH-2E core contribute to higher speeds of execution and greater code efficiency.

On-chip peripheral functions essential to system configuration on this LSI include a floating point unit (FPU), large-capacity ROM and RAM, a direct memory access controller (DMAC), timers, an automotive direct memory access controller (A-DMAC), a Renesas serial peripheral interface (RSPI), an user break controller (UBC), an advanced user debugger II (AUD-II), an advanced timer unit III (ATU-III), a serial communications interface (SCI), a controller area network interface (RCAN-TL1), an A/D converter (ADC), an interrupt controller (INTC), and I/O ports.

This LSI also includes functions for external access control that allow the direct connection of various external memory modules and peripheral devices, and has the potential to greatly reduce system costs.

The on-chip ROM of this LSI takes the form of flexible zero turn-around time (F-ZTAT) memory, i.e. flash memory. Flash memory can be reprogrammed and erased by software. This allows the user to reprogram the chip while it is mounted on the board.

The features of this LSI are listed in table 1.1.

Note: F-ZTAT is a trademark of Renesas Electronics Corporation.

Section 1 Overview SH72546R

#### Table 1.1 SH72546R Features

# Item **Features CPU** Original Renesas SuperH architecture Upwardly compatible with SH-1, SH-2, and SH-2E cores on the object code level 32-bit internal data bus General-register architecture Sixteen 32-bit general registers Four 32-bit control registers Four 32-bit system registers Register banks for fast interrupt response RISC-type instruction set (upward-compatible with SH-2E Series) Instruction length: 16-bit basic instructions for improved code efficiency, and 32-bit instructions for improved performance and ease of use Load-store architecture Delayed branch instructions Instruction set based on C language Superscalar architecture allowing simultaneous execution of two instructions, including FPU Instruction execution time: Max. 2 instructions/cycle Address space: 4 Gbytes On-chip multiplier Five-stage pipeline

Harvard architecture

SH72546R Section 1 Overview

**Features** Item Floating-point unit On-chip floating-point coprocessor (FPU) Supports single-precision (32 bits) and double-precision (64 bits) Supports IEEE 754-compliant data types and exceptions Two rounding modes: Round to Nearest and Round to Zero Handling of denormalize numbers: Truncation to zero Floating-point registers Sixteen 32-bit floating-point registers (single-precision x 16 words or double-precision x 8 words) Two 32-bit floating-point system registers Supports FMAC (multiply and accumulate) instruction Supports FDIV (division) and FSQRT (square root) instructions Supports FLDI0/FLDI1 (load constant 0/1) instructions Instruction execution times Latency (FMAC/FADD/FSUB/FMUL): 3 cycles (single-precision), 8cycles (double-precision) Pitch (FMAC/FADD/FSUB/FMUL): 1 cycle (single-precision), 6cycles (double-precision) Note: FMAC is supported for single-precision only. Five-stage pipeline

Selectable exception generation when qNaN or  $\pm \infty$  input

Section 1 Overview SH72546R

| Item                 | Features  |
|----------------------|---|
| Operating modes      | Operating modes   |
|                      | Single-chip mode  |
|                      | 8/16-bit bus extended mode  |
|                      | — On-chip ROM enabled   |
|                      | On-chip ROM disabled  |
|                      | On-board programming mode   |
|                      | Boot mode   |
|                      | User boot mode  |
|                      | User program mode   |
|                      | Processing states   |
|                      | Reset state   |
|                      | Program execution state   |
|                      | Exception handling state  |
|                      | Power-down modes  |
|                      | Sleep mode  |
|                      | Hardware standby mode   |
| _                    | Module standby mode   |
| Clock pulse          | On-chip oscillation circuit (Maximum operating frequency: 200 MHz)  |
| generator (CPG)      | Two types of clock generation   |
|                      | Internal clock (200 MHz max)  |
|                      | Peripheral clock (40 MHz max)   |
|                      | Internal/peripheral clocks multiplied by the on-chip PLL  |
|                      | $\times 4, \times 6, \times 8,$ or $\times 10$ can be selected for the internal clock multiplication by setting the MD_CLK0/1 pins. |
|                      | $\times 1$ or $\times 2$ can be selected for the peripheral clock multiplication by setting the MD_CLKP pin.                        |
|                      | Input clock frequency: 16 to 20 MHz   |
| Interrupt controller | Nine external interrupt pins (NMI and IRQ0 to IRQ7)   |
| (INTC)               | Fifteen software interrupts with priority levels 1 to 15  |
|                      | Sixteen programmable priority levels  |

| Item                       | Features   |
|----------------------------|--|
| User break                 | 4 channels   |
| controller (UBC)           | Generates an interrupt when the CPU, DMAC, or A-DMAC generates a   |
|                            | bus cycle with specified conditions. (Interrupt can be masked.)  |
|                            | Generates a trigger pulse output (UBCTRG) on satisfaction of a break   |
|                            | condition (the pulse width is selectable as 1, 2, 4, or 8 P $\phi$ cycles)   |
|                            | Pulse width of 1 P $\phi$ cycle is only available when $\times 1$ multiplication has been set for the peripheral clock |
| Bus state controller (BSC) | <ul> <li>Supports external memory accesses (SRAM and ROM directly connectable)</li> </ul>                              |
|                            | • 8/16-bit bus   |
|                            | 3.3-V bus interface  |
|                            | 16-Mbyte address space divided into four areas, with the following   |
|                            | parameters settable for each area:   |
|                            | Bus size: 8 or 16 bits   |
|                            | Number of wait cycles  |
|                            | Chip select signals ( $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ ) output for each area                        |
|                            | Wait cycle insertion with external WAIT signal   |
|                            | Provision for idle cycle insertion to prevent bus conflicts  |
| Direct memory              | Eight channels   |
| access controller (DMAC)   | DMA transfer possible between following devices/modules:   |
| (DIVI/NO)                  | external memory, on-chip memory, on-chip peripheral modules (excluding DMAC and A-DMAC)                                |
|                            | Cycle stealing or burst mode transfer  |
|                            | Dual address mode  |
|                            | Reloading function   |
|                            | Reloading source or/and destination address and reload counter   |
|                            | Reloading source and destination address, and transfer and reload counter  |
|                            | $\bullet  \text{Integer} \rightarrow \text{floating-point conversion selection during transfer}$                       |
|                            | Transfer data width: byte/word/longword/16 bytes   |

### **Features** Item Automotive direct 66 channels memory access Forwarding data from/to the specific module to/from alias (specific RAM controller (Aaddress) is possible triggered by event generation DMAC) Following modules are supported One channel for ADC: A/D converted value transfer One channel for ATU-III (timer G): RAM data transfer from RAM to PORT Channels for ATU-III (timers A, C, and F): Input capture value transfer for timer A (6 channels), timer C (20 channels), and timer F (20 channels) Six channels for RSPI: MISOA to MISOC and MOSIA to MOSIC Ten channels for SCI: RxD A to RxD E and TxD A to TxD E Two channels for RCAN-TL1: CRx\_A to CRx\_C and CTx\_A to CTx\_C Alias address pointers One base pointer: Addresses of aliases for ATU-III (timers A, C, and F) and RCAN-TL1 channels relative to the base address are fixed. 18 alias pointers: One for each of ATU-III (timer G), ADC, RSPI, and SCI channels Advanced timer Timer A: 32-bit input capture inputs $\times$ 6 channels unit (ATU-III) Timer B: Angle clock generation timer × 1 channel Timer C: 24-bit input capture/output compare × 20 channels Timer D: 24-bit one shot pulse × 16 channels 24-bit output compare × 16 channels Timer E: 16-bit PWM × 24 channels Timer F: 24-bit event counter × 20 channels Timer G: 16-bit interval timer × 6 channels Timer H: 32-bit interval timer × 1 channel Timer J: 16-bit input capture with 9-stage FIFO × 2 channels Two external clocks can be input as clock source External input pins with noise canceller (timer A, C, F, and J) Watchdog timer Selectable from watchdog timer or interval timer function (WDT) Internal reset, external signal, or interrupt generated by counter overflow Power-on reset

| Item                           | Features   |
|--------------------------------|--|
| Compare match                  | Two channels   |
| timer (CMT)                    | Selection of four counter-input clocks   |
|                                | <ul> <li>A compare-match interrupt can be requested independently for each channel</li> </ul>                        |
| Serial                         | Five channels  |
| communications interface (SCI) | Selection of asynchronous or synchronous mode  |
| interface (SOI)                | Simultaneous transmission/reception (full-duplex) capability   |
|                                | Maximum baud rate  |
|                                | Asynchronous: 1.25 Mbps  |
|                                | Synchronous: 2.5 Mbps  |
| Renesas serial                 | Three channels   |
| peripheral interface (RSPI)    | Synchronous serial communications  |
| interface (Fier i)             | Master/slave mode supported  |
|                                | Programmable bit length, clock polarity, and clock phase   |
|                                | Sequential loop transfer capable   |
|                                | MSB/LSB first selectable   |
|                                | Maximum transfer rate: 10 MHz  |
|                                | <ul> <li>Channel A can control up to 8 slaves in single master mode (depends on<br/>PFC settings).</li> </ul>        |
|                                | Channel A can control up to 7 slaves in multi master mode (depends on PFC settings).                                 |
|                                | <ul> <li>Channels B and C can control up to 4 slaves in single master mode<br/>(depends on PFC settings).</li> </ul> |
|                                | Channels B and C can control up to 3 slaves in multi-master mode (depends on PFC settings).                          |

| Item                  | Features  |
|-----------------------|---|
| Controller area       | Three channels  |
| network<br>(RCAN-TL1) | TTCAN level 1 support for all channels  |
| (HOAN-TET)            | BOSCH 2.0B active compatible  |
|                       | Buffer size: transmit/receive × 31, receive only × 1  |
|                       | Two or more RCAN-TL1 channels can be assigned to one bus to increase<br>number of buffers with a granularity of 32 channels |
|                       | <ul> <li>Parity error detection capability (1-bit parity based; every 8-bit message<br/>buffer)</li> </ul>                  |
|                       | Capability of interrupt generation upon detection of a parity error   |
| A/D converter         | 37 channels   |
| (ADC)                 | Two sample-and-hold circuits  |
|                       | Independent operation of 12-bit 28 channels and 9 channels  |
|                       | Selectable from three conversion modes  |
|                       | Continuous scan mode  |
|                       | Single scan mode  |
|                       | A/D conversion value addition mode (a single channel is converted continuously two to four times)                           |
|                       | Conversion trigger capability   |
|                       | Scanning can be started by external trigger or ATU-III compare-match  |
|                       | Self-test capability  A Worf and A Worf and the Rep (4.40 LOB) and the management of  |
|                       | <ul> <li>0 × Avref, 0.5 × AVref, or 1 × AVref voltage (±40 LSB) can be generated internally.</li> </ul>                     |
|                       | Conversion time:  |
|                       | 25/50 P $\phi$ cycles when the peripheral clock multiplication ratio is set to $\times 1$                                   |
|                       | 50 P $\phi$ cycles when the peripheral clock multiplication ratio is set to $\times 2$                                      |
|                       | <ul> <li>Accuracy: ±8 LSB, Non linearity error: ±4 LSB</li> </ul>   |
| JTAG interface        | JTAG port   |
|                       | Boundary scan test ports supporting IEEE 1149.1   |

| Item   | Features   |
|--|--|
| Advanced user                                | Eight dedicated pins   |
| debugger II<br>(AUD-II)                      | RAM monitor mode   |
| (AOD-II)                                     | Data input/output frequency: equal to or lower than both the peripheral clock frequency (P $\phi$ ) and 20 MHz |
|  | Possible to read from or write to a module connected to the internal/external bus                              |
|  | AUD trace mode   |
| I/O ports                                    | Selectable output driving ability for specific I/O pins  |
|  | <ul> <li>Output inversion enabled or disabled for specific I/O pins</li> </ul>                                 |
|  | Switchable pull-down resistors for the MISO and RxD pins   |
|  | CK pin output enabled or disabled by register setting  |
|  | Edge detection registers provided for specific ports   |
| Multi-input<br>signature<br>generator (MISG) | Monitors CPU write accesses to specific addresses and generates 32-bit signature by the written data           |
| ROM  | 3.75-Mbyte flash memory  |
|  | ROM cache  |
|  | Instruction cache: full associative, 8 lines, 16 byte/line   |
|  | Data cache: full associative, 4 lines, 16 byte/line  |
|  | Line size: 16 byte/line  |
| EEPROM                                       | 128-Kbyte EEPROM   |
|  | Sixteen 8-Kbyte blocks   |
|  | Writing is possible in 8-byte units, erasure is possible in 8-Kbyte units                                      |
| RAM  | 256-Kbyte SRAM   |
|  | ECC: 1-bit correction and 2-bit detection  |
|  | ECC can be enabled or disabled   |
| Package                                      | • 272-pin BGA  |

# 1.2 Block Diagram

Figure 1.1 shows a block diagram of the SH72546R.

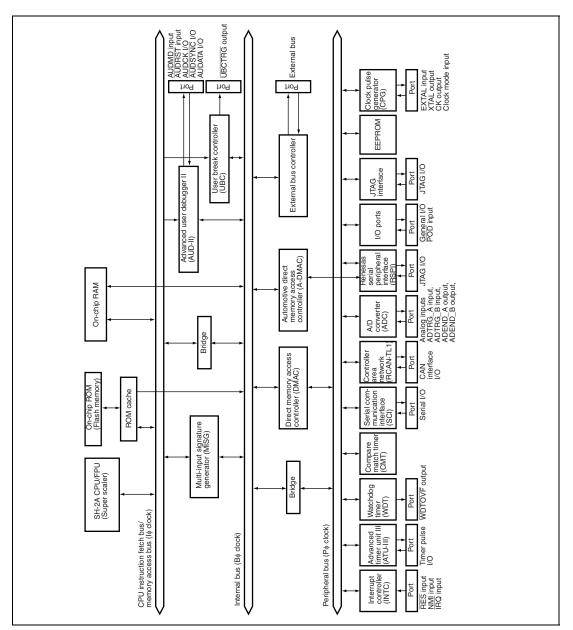


Figure 1.1 Block Diagram

# 1.3 Pin Descriptions

### 1.3.1 Pin Arrangements

The pin arrangements of this LSI is shown in figure 1.2.

| 7 | 1                                   | 2                                   | 3                                  | 4                                   | 5                                   | 6                                   | 7                        | - 8                                 | 9                                   | 10                                  | 11                      | 12                       | 13                        | 14                        | 15                                 | 16  | 17  | 18  | 19                                  | 20                             |
|---|-------------------------------------|-------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------|--------------------------|---------------------------|---------------------------|------------------------------------|---|---|---|-------------------------------------|--------------------------------|
|   | Vss                                 | PE10/<br>TOE10                      | PE11/<br>TOE11                     | PD2/<br>TIOC02/<br>TOE21/<br>TOE52  | PD3/<br>TIOC03/<br>TOE22/<br>TOE53  | PD6/<br>TIOC12/<br>TOE41            | PD7/<br>TIOC13/<br>TOE42 | PD8/<br>TIOC20/<br>TIOC33/<br>TOE53 | PD11/<br>TIOC23/<br>TIF2B/<br>TOE51 | Vss                                 | FWE                     | MD1                      | ск                        | EXTAL                     | XTAL                               | PLLVss  | PLLVcc  | ASEMD                                       | TDI                                 | Vss                            |
| 3 | PH0/<br>ADTRG_A/<br>TIF0A           | PH3/<br>TIF3                        | PE5/<br>TIA05                      | PE6/<br>TOE00/<br>CTx_B             | PE8/<br>TOE02                       | PD0/<br>TIOC00/<br>TIOC31           | PD1/<br>TIOC01/<br>TOE20 | PD5/<br>TIOC11/<br>TOE23/<br>TOE40  | PD9/<br>TIOC21/<br>TIF0B/<br>TOE43  | VCL                                 | WDTOVF                  | RES                      | Vss                       | Voc                       | MD_CLKP                            | MD_CLK1   | TRST  | тск   | PA1/<br>A1                          | PA7/<br>A7                     |
| , | PK7/<br>TxD_E                       | PK11/<br>MISOC                      | PH5/<br>TIF5                       | PE4/<br>TIA04                       | PE3/<br>TIA03                       | PE2/<br>TIA02/<br>TIOC43/<br>TIOC30 | PE9/<br>TOE03            | PE12/<br>TOE12                      | PD4/<br>TIOC10/<br>TIOC32/<br>TOE52 | PD10/<br>TIOC22/<br>TIF1B/<br>TOE50 | PD13/<br>TCLKB/<br>TIJ1 | MD0                      | MD3                       | MD4                       | NMI                                | TDO   | AUDMD   | PAO/<br>AO                                  | PA2/<br>A2                          | PA9/<br>A9                     |
| 0 | PK6/<br>SCK_E                       | PK10/<br>MOSIC                      | PH4/<br>TIF4                       | PE0/<br>TIA00                       | PE1/<br>TIA01/<br>TIOC42/<br>TIOC40 | PE7/<br>TOE01/<br>CRx_B             | Vcc                      | PE13/<br>TOE13                      | PVcc2                               | PD12/<br>TCLKA/<br>TIOC41/<br>TIJ0  | HSTBY                   | MD2                      | Voc                       | MD_CLK0                   | TMS                                | AUDRST  | AUDATA3                                       | AUDSYNC                                     | PA4/<br>A4                          | PA12/<br>A12                   |
|   | PK3/<br>SCK_D/<br>RSPCKB            | PK8/<br>RxD_E                       | PH1/<br>ADTRG_B/<br>TIF1A          | PVcc2                               |                                     |                                     |                          |                                     |                                     |                                     |                         |                          |                           |                           |                                    |   | AUDATA1                                       | AUDATA0                                     | PA3/<br>A3                          | PA13/<br>A13                   |
|   | PK2/<br>RxD_C/<br>MISOA             | PK5/<br>RxD_D/<br>MISOB             | PK9/<br>RSPCKC                     | PH2/<br>TIF2A                       |                                     |                                     |                          |                                     |                                     |                                     |                         |                          |                           |                           |                                    |   | Voc   | AUDATA2                                     | PA6/<br>A6                          | PA15/<br>A15                   |
| à | PG0/<br>TOD00A/<br>SSLA0            | PK0/<br>SCK_C/<br>RSPCKA/<br>UBCTRG | PK4/<br>TxD_D/<br>MOSIB            | Vec                                 |                                     |                                     |                          |                                     |                                     |                                     |                         |                          |                           |                           |                                    |   | PA5/<br>A5                                    | AUDCK                                       | PA11/<br>A11                        | PB0/<br>A16/<br>MOSIA          |
| 1 | PG3/<br>TOD03A/<br>SSLA3            | PG2/<br>TOD02A/<br>SSLA2            | PG1/<br>TOD01A/<br>SSLA1           | PK1/<br>TxD_C/<br>MOSIA             |                                     |                                     |                          |                                     |                                     |                                     |                         |                          |                           |                           |                                    |   | PVcc1   | PAB/<br>A8                                  | PA14/<br>A14                        | PB4/<br>A20/<br>CTx_B/<br>TIF6 |
| J | PG4/<br>TOD10A/<br>SSLA4/<br>SSLB3  | PG6/<br>TOD12A/<br>SSLB0            | PG5/<br>TOD11A/<br>SSLA5/<br>SSLC3 | PVcc2                               |                                     |                                     |                          |                                     | Vss                                 | Vss                                 | Vsa                     | Vss                      |                           |                           |                                    |   | PVcc1   | PA10/<br>A10                                | PB2/<br>A18/<br>MOSIB               | PB5/<br>A21/<br>CRx_B/<br>TIF7 |
| < | Vss                                 | VCL                                 | PG9/<br>TOD21A/<br>SSLC0/<br>TIF7  | PG7/<br>TOD13A/<br>SSLB1            |                                     |                                     |                          |                                     | Vss                                 | Vss                                 | Vss                     | Vss                      |                           |                           |                                    |   | PVoc1   | PB1/<br>A17/<br>MISOA                       | PB6/<br>WE0                         | PB8/<br>WAIT/<br>TOE20         |
| 1 | PG8/<br>TOD20A/<br>SSLB2/<br>TIF6   | PG10/<br>TOD22A/<br>SSLC1/<br>TIF8  | PG11/<br>TOD23A/<br>SSLC2/<br>TIF9 | Voc                                 |                                     |                                     |                          |                                     | Vss                                 | Vss                                 | Vss                     | Vss                      |                           |                           |                                    |   | PB7/<br>WE1                                   | PB3/<br>A19/<br>MISOB                       | VCL                                 | Vss                            |
| Л | PG12/<br>TOD30A/<br>SSLA4/<br>TIF10 | PG14/<br>TOD32A/<br>SSLA6/<br>TIF12 | PL7/<br>TOE32/<br>IRQ7             | PG13/<br>TOD31A/<br>SSLA5/<br>TIF11 |                                     |                                     |                          |                                     | Vss                                 | Vss                                 | Vss                     | Vss                      |                           |                           |                                    |   | Voc   | PB10/<br>CS0                                | PB9/<br>RD                          | PB11/<br>CS1/<br>TOE21         |
| ٧ | PG15/<br>TOD33A/<br>SSLA7/<br>TIF13 | PL6/<br>TOE31/<br>IRQ6              | PL2/<br>TOE21/<br>IRQ2             | PVcc2                               |                                     |                                     |                          |                                     |                                     |                                     |                         |                          |                           |                           |                                    |   | PC3/<br>D3                                    | PC0/<br>D0                                  | PB13/<br>CS3/<br>RSPCKB             | PB12/<br>CS2/<br>RSPCKA        |
| , | PL8/<br>TOE33                       | PL4/<br>TOE23/<br>IRQ4              | PL0/IRQ0                           | PL1/<br>TOE20/<br>IRQ1/<br>POD      | 1                                   |                                     |                          |                                     |                                     |                                     |                         |                          |                           |                           |                                    |   | PVcc1   | PC7/<br>D7                                  | PC2/<br>D2                          | PB14/<br>RD/WR                 |
| 2 | PL5/<br>TOE30/<br>IRQ5              | PL3/<br>TOE22/<br>IRQ3              | AN_B44                             | AN_B46                              |                                     |                                     |                          |                                     |                                     |                                     |                         |                          |                           |                           |                                    |   | PVcc1   | PC10/<br>D10                                | PC5/<br>D5                          | PC1/<br>D1                     |
| г | AVcc                                | AN_B48                              | AN_B40                             | AN_B43                              |                                     |                                     |                          |                                     |                                     |                                     |                         |                          |                           |                           |                                    |   | PJ4/<br>SCK_A/<br>ADEND_B/<br>TIJ0            | PC15/<br>D15                                | PC6/<br>D6                          | PC4/<br>D4                     |
| J | AVss                                | AN_B47                              | AN_B42                             | AN_B41                              | AN_A23                              | AN_A21                              | AN_A19                   | Voc                                 | AN_A8                               | AN_A4                               | AN_A1                   | PVcc2                    | PF7/<br>TOD13B/<br>TIF13  | Voc                       | PJ7/<br>SCK_B/<br>ADEND_A/<br>TIJ1 | PVcc2   | PJ3/<br>RxD_A/<br>CRx_C/<br>CRx_A&CRs_B&CRx_C | PC14/<br>D14                                | PC9/<br>D9                          | PC8/<br>D8                     |
| 7 | AVREFL_B                            | AN_B45                              | AN_A26                             | AN_A27                              | AN_A22                              | AN_A20                              | AN_A15                   | AN_A13                              | AN_A6                               | AN_A3                               | PF0/<br>TOD00B/<br>TIF6 | PF4/<br>TOD10B/<br>TIF10 | PF10/<br>TOD22B/<br>TIF16 | PF12/<br>TOD30B/<br>TIF18 | PJ5/<br>TxD_A                      | PJ2/<br>TxD_A/<br>CTx_C/<br>CTx_A&CTx<br>_B&CTx_C | PJ1/<br>RxD_A/<br>CRx_A/<br>CRx_A&<br>CRx_B   | PJ0/<br>TxD_A/<br>CTx_A/<br>CTx_A&<br>CTx_B | PC13/<br>D13                        | PC11/<br>D11                   |
| ٧ | AVREFH_B                            | AN_A25                              | AVREFL_A                           | AVss                                | AN_A18                              | AN_A17                              | AN_A16                   | AN_A12                              | AN_A7                               | AVcc                                | AN_A0                   | VCL                      | PF3/<br>TOD03B/<br>TIF9   | PF8/<br>TOD20B/<br>TIF14  | PF13/<br>TOD31B/<br>TIF19          | PF15/<br>TOD33B/<br>CRx_B/<br>RxD_A               | PJ9/<br>RxD_B                                 | PJ8/<br>TxD_B                               | PJ6/<br>RxD_A                       | PC12/<br>D12                   |
| 1 | NC                                  | AN_A24                              | AVREFH_A                           | AVcc                                | AN_A14                              | AN_A11                              | AN_A10                   | AN_A9                               | AN_A5                               | AVss                                | AN_A2                   | Vss                      | PF1/<br>TOD01B/<br>TIF7   | PF2/<br>TOD02B/<br>TIF8   | PF5/<br>TOD11B/<br>TIF11           | PF6/<br>TOD12B/<br>TIF12                          | PF9/<br>TOD21B/<br>TIF15                      | PF11/<br>TOD23B/<br>TIF17                   | PF14/<br>TOD32B/<br>CTx_B/<br>TxD_A | Vss                            |
| 1 | 1                                   | 2                                   | 3                                  | 4                                   | 5                                   | 6                                   | 7                        | 8                                   | 9                                   | 10                                  | 11                      | 12                       | 13                        | 14                        | 15                                 | 16  | 17  | 18  | 19                                  | 20                             |

Figure 1.2 Pin Arrangements

### 1.3.2 Pin Functions

Table 1.2 lists pin functions.

**Table 1.2 Pin Functions** 

| Classi-<br>fication | Symbol             | Pin No. I/O                        | Name   | Function   |
|---------------------|--------------------|------------------------------------|--|--|
| Power supply        | V <sub>cc</sub>    | B14, D7, I<br>D13, F17,<br>G4, L4, | Power supply                                       | Power supply pins for internal and system pins. All the $V_{\rm cc}$ pins must be connected to the system power supply.  |
|                     |                    | M17, U8,<br>U14                    |  | This LSI does not operate correctly if there is a pin left open.   |
|                     | PV <sub>cc</sub> 1 | H17, J17,   l<br>K17, P17,<br>R17  | 5-V/3.3-V<br>power supply<br>(for extended<br>bus) | Power supply for bus ports (ports A, B, and C). Connect all PV <sub>cc</sub> 1 pins to the system bus power supply. This LSI does not operate correctly if there is a pin left open.   |
|                     | PV <sub>cc</sub> 2 | D9, E4, I<br>J4, N4,               | 5-V power supply                                   | Power supply for peripheral module ports (ports D, E, F, G, H, J, K, and L).   |
|                     |                    | U12, U16                           |  | Connect all $PV_{cc}2$ pins to the system peripheral module power supply. This LSI does not operate correctly if there is a pin left open.   |
|                     | V <sub>cL</sub>    | B10, K2, I<br>L19, W12             | Internal step-<br>down power<br>supply             | Pins for connection to a capacitor used for stabilizing the voltage of the internal step-down power supply. Connect V <sub>ss</sub> to this pin through a capacitor. The capacitor should be located near the pin. Do not connect an external power supply to the pin. |

| Classi-<br>fication    | Symbol                          | Pin No.  | I/O | Name                  | Function   |
|------------------------|---------------------------------|--|-----|-----------------------|--|
| Power<br>supply        | V <sub>ss</sub>                 | A1, A10,<br>A20, B13,<br>J9, J10,<br>J11, J12,<br>K1, K9,<br>K10, K11,<br>K12, L9,<br>L10, L11,<br>L12, L20,<br>M9, M10,<br>M11,<br>M12,<br>Y12, Y20 |     | Ground                | Ground pins. All the $V_{\rm ss}$ pins must be connected to the system power supply (0V). This LSI does not operate correctly if there is a pin left open. |
| Clocks                 | PLLV <sub>cc</sub>              | A17  | I   | PLL power supply      | Power supply for the on-chip PLL oscillator.   |
|                        | PLLV <sub>ss</sub>              | A16  | I   | PLL ground            | Ground pin for the on-chip PLL oscillator.   |
|                        | EXTAL                           | A14  | I   | External clock        | Connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin.   |
|                        | XTAL                            | A15  | 0   | Crystal               | Connected to a crystal resonator.  |
|                        | СК                              | A13  | 0   | Peripheral clock      | Supplies the peripheral clock to peripheral devices.   |
| Operating mode control | ASEMD                           | A18  | I   | ASE mode              | Enables emulator functions. Input a low level to operate the LSI in normal mode (other than debugging mode).   |
|                        |                                 |  |     |                       | In debugging mode, input a high level to this pin on the user system board.  |
|                        | FWE                             | A11  | I   | Flash                 | Input a low level in normal operation.   |
|                        |                                 |  |     | programming<br>enable | Input a high level during on-board programming.  |
|                        | MD4 to MD0                      | C12,<br>A12,<br>D12,<br>C13, C14   | I   | Mode set              | Sets the operating mode. Do not change signal levels on these pins during operation.   |
|                        | MD_CLK1,<br>MD_CLK0,<br>MD_CLKP | D14,<br>B15, B16   | I   | Clock mode set        | Sets the clock operating mode. Do not change signal levels on these pins during operation.   |

| Classi-<br>fication | Symbol       | Pin No.   | I/O | Name                            | Function   |
|---------------------|--------------|---|-----|---------------------------------|--|
| System control      | HSTBY        | D11   | 1   | Hardware standby                | Driving this pin low puts the LSI into hardware standby mode.  |
|                     | RES          | B12   | I   | Power-on reset                  | Driving this pin low puts the LSI into power-on reset state.   |
|                     | WDTOVF       | B11   | 0   | Watchdog<br>timer overflow      | WDT overflow output signal   |
| Interrupts          | NMI          | C15   | I   | Non-<br>maskable<br>interrupt   | Non-maskable interrupt request pin   |
|                     | IRQ7 to IRQ0 | P3, P4,<br>N3, R2,<br>P2, R1,<br>N2, M3   | I   | Interrupt<br>requests<br>7 to 0 | Maskable interrupt request pins. Level-<br>input or edge-input detection can be<br>selected. When the edge input detection is<br>selected, the rising edge, falling edge, or<br>both edges can also be selected. |
| Address bus         | A21 to A0    | C18, B19 C19, E19, D19, G17, F19, B20, H18, C20, J18, G19, D20, E20, H19, F20, G20, K18, J19, L18, H20, J20 |     | Address bus                     | Address bus  |

| Classi-<br>fication    | Symbol            | Pin No.  | I/O | Name                                  | Function   |
|------------------------|-------------------|--|-----|---------------------------------------|--|
| Data bus               | D15 to D0         | N18, R20<br>P19,<br>N17, T20,<br>R19, T19,<br>P18,<br>U20,<br>U19,<br>R18,<br>V20,<br>W20,<br>V19,<br>U18, T18 |     | Data bus                              | Bidirectional data bus   |
| Bus control            | CS3 to CS0        | M18,<br>M20,<br>N20, N19   | 0   | Chip select<br>3 to 0                 | Chip-select signals for external memory or devices   |
|                        | RD                | M19  | 0   | Read                                  | Indicates that data is read from an external device.   |
|                        | RD/WR             | P20  | 0   | Read/write                            | Read/write signal.   |
|                        | WAIT              | K20  | I   | Wait control                          | Input signal for inserting a wait cycle into the bus cycles during access to the external space. |
|                        | WE0               | K19  | 0   | Write/byte select                     | Indicates a write access to bits 7 to 0 of data bus of external memory or device.                |
|                        | WE1               | L17  | 0   | Write/byte select                     | Indicates a write access to bits 15 to 8 of data bus of external memory or device.               |
| Advanced timer unit II | TCLKA, TCLKB      | D10, C11   | I   | ATU-III timer clock input             | Input pins for the ATU-III counter external clock  |
| (ATU-III)              | TIA05 to TIA00    | D4, D5,<br>C6, C5,<br>C4, B3   | I   | ATU-III input<br>capture<br>(timer A) | Timer A input capture input pins   |
|                        | TIOC43 to TIOC40, | B6, B7,  | I/O | ATU-III input                         | Timer C input capture input/output   |
|                        | TIOC33 to TIOC30, | C10, A9,   |     | capture/                              | compare output pins  |
|                        | TIOC23 to TIOC20, | D10, A4,<br>A5, C9,  |     | output<br>compare                     |  |
|                        | TIOC13 to TIOC10, | B8, A6,  |     | (timer C)                             |  |
|                        | TIOC03 to TIOC00  | A7, A8,<br>B9, D5,<br>C6   |     |                                       |  |

| Classi-<br>fication                    | Symbol   | Pin No.  | I/O | Name  | Function  |
|--|--|--|-----|---|---|
| Advanced<br>timer unit II<br>(ATU-III) | TOD33A to TOD30A,<br>TOD23A to TOD20A,<br>TOD13A to TOD10A,<br>TOD03A to TOD00A                | L2, L3,<br>M1, M4,   | 0   | ATU-III<br>compare<br>match<br>(timer D)        | Timer D output compare match output pins        |
|  | TOD33B to TOD30B, TOD23B to TOD20B, TOD13B to TOD10B, TOD03B to TOD00B                         | V13, Y18,  | 0   | ATU-III<br>one-shot<br>pulse<br>(timer D)       | Timer D down-counter one-shot pulse output pins |
|  | TOE53 to TOE50, TOE43 to TOE40, TOE33 to TOE30, TOE23 to TOE20, TOE13 to TOE10, TOE03 to TOE00 | B4, D6,<br>B5, C7,<br>A2, A3,<br>C8, D8,<br>P4, K20,<br>B7, N3,<br>M20, A4<br>R2, A5,<br>P2, B8,<br>R1, N2,<br>M3, P1,<br>A6, A7,<br>B9, C10,<br>A9, C9, | 0   | ATU-III<br>PWM output<br>(timer E)              | Timer E output compare/PWM output pins          |
|  | TIF2A to TIF0A   | B1, E3,<br>F4  | I   | ATU-III event<br>input<br>(phase A,<br>timer F) | Timer F event input pins (phase A)              |

| Classi-<br>fication                       | Symbol         | Pin No.   | I/O | Name  | Function                                 |
|---|----------------|---|-----|---|--|
| Advanced<br>timer unit II<br>(ATU-III)    | TIF2B to TIF0B | B9, C10,<br>A9  | I   | ATU-III event<br>input<br>(phase B,<br>timer F) | Timer F event input pins (phase B)       |
|   | TIF19 to TIF3  | V12, M1,<br>Y15, M2,<br>U13, N1,<br>W14,<br>Y17, V13,<br>Y8, V14,<br>W15, B2,<br>D3, C3,<br>V11,<br>H20, L1,<br>Y13, J20,<br>K3, Y14,<br>L2, W13,<br>L3 |     | ATU-III event input (timer F)                   | Timer F event input pins                 |
|   | TIJ1, TIJ0     | D10, T17,<br>C11, U15   | I   | ATU-III<br>capture input<br>(timer J)           | Timer J input capture input pins         |
| Controller<br>area<br>network             | CTx_A to CTx_C | V18, V16,<br>H20, B4,<br>Y19  | 0   | Transmit data<br>(channels<br>A to C)           | CAN bus transmit data output pins        |
| (RCAN-TL1)                                | CRx_A to CRx_C | V17,<br>U17, J20,<br>D6, W16  | I   | Receive data<br>(channels<br>A to C)            | CAN bus receive data input pins          |
| Serial<br>communi-<br>cation<br>interface | TxD_A to TxD_E | V18, V16,<br>V15, Y19,<br>W18, H4,<br>G3, C1  |     | Transmit data<br>(channels<br>A to E)           | SCI_A to SCI_E transmit data output pins |
| (SCI)                                     | RxD_A to RxD_E | V17,<br>U17,<br>W19,<br>W16,<br>W17, F1,  | ı   | Receive data<br>(channels<br>A to E)            | SCI_A to SCI_E receive data input pins   |
|   | SCK_A to SCK_E | T17, U15,<br>G2, E1,<br>D1  | I/O | Serial clock<br>(channels<br>A to E)            | SCI_A to SCI_E clock input/output pins   |

| Classi-<br>fication               | Symbol                       | Pin No.  | I/O | Name                                  | Function   |
|-----------------------------------|------------------------------|--|-----|---------------------------------------|--|
| Serial<br>peripheral<br>interface | MOSIA to MOSIC               | G20, H4,<br>J19, G3,<br>D2   | I/O | Transmit data<br>(channels<br>A to C) | RSPI_A to RSPI_C transmit data input/output pins |
| (RSPI)                            | MISOA to MISOC               | K18, F1,<br>L18, F2,<br>C2   | I/O | Receive data<br>(channels<br>A to C)  | RSPI_A to RSPI_C receive data input/output pins  |
|                                   | RSPCKA to<br>RSPCKC          | N20, G2,<br>N19, E1,<br>F3   | I/O | Serial clock<br>(channels<br>A to C)  | RSPI_A to RSPI_C clock input/output pins         |
|                                   | SSLA0, SSLB0,<br>SSLC0       | G1, J2,<br>K3  | I/O | Chip select<br>(channels<br>A to C)   | RSPI_A to RSPI_C chip select input/output pins   |
|                                   | SSLA1-7, SSLB1-3,<br>SSLC1-3 | H3, H2,<br>H1, M1,<br>J1, M4,<br>J3, M2,<br>N1, K4,<br>L1, L2,<br>L3 | 0   | Chip select<br>(channels<br>A to C)   | RSPI_A to RSPI_C chip select input/output pins   |
| User break<br>controller<br>(UBC) | UBCTRG                       | G2   | 0   | User break<br>trigger output          | Trigger output pin for UBC condition match       |
| A/D<br>converter                  | AV <sub>cc</sub>             | T1, W10,<br>Y4   | ı   | Analog power supply                   | Power supply pins for the A/D converter          |
| (ADC)                             | AV <sub>ss</sub>             | U1, W4,<br>Y10   | I   | Analog<br>ground                      | Ground pins for the A/D converter                |

| Classi-<br>fication       | Symbol                               | Pin No.  | I/O | Name                                | Function  |
|---------------------------|--------------------------------------|--|-----|-------------------------------------|---|
| A/D<br>converter<br>(ADC) | AN_A27 to AN_A0,<br>AN_B48 to AN_B40 | W11,<br>U11, Y7,<br>Y6, W8,<br>V8, Y5,<br>V7, W7,<br>W6, W5,<br>U7, Y11,<br>V6, U6,<br>V5, U5,<br>Y2, W2,<br>V3, V4,<br>V10,<br>U10, Y9,<br>V9, W9,<br>U9, Y8,<br>T3, U4,<br>U3, T4,<br>R3, V2,<br>R4, U2,<br>T2 | ı   | Analog input pins                   | Analog input pins                                       |
|                           | ADTRG_A,<br>ADTRG_B                  | B1, E3   | I   | A/D<br>conversion<br>trigger input  | External trigger input pin for starting A/D conversion  |
|                           | ADEND_A,<br>ADEND_B                  | U15, T17   | 0   | ADEND<br>output                     | Output pins for A/D conversion timing monitor           |
|                           | AVREFH_A,<br>AVREFH_B                | Y3, W1   | I   | Analog<br>reference<br>power supply | Input pins for analog high level reference power supply |
|                           | AVREFL_A,<br>AVREFL_B                | W3, V1   | I   | Analog<br>reference<br>power supply | Input pins for analog low level reference power supply  |
| JTAG                      | TCK                                  | B18  | I   | Test clock                          | Test-clock input pin                                    |
| interface                 | TMS                                  | D15  | I   | Test mode select                    | Test-mode select signal input pin                       |
|                           | TDI                                  | A19  | I   | Test data input                     | Serial input pin for instructions and data              |
|                           | TDO                                  | C16  | 0   | Test data output                    | Serial output pin for instructions and data             |
|                           | TRST                                 | B17  | I   | Test reset                          | Initialization-signal input pin                         |

| Classi-<br>fication     | Symbol                | Pin No.               | I/O | Name                | Function  |
|-------------------------|-----------------------|-----------------------|-----|---------------------|---|
| Advanced user           | AUDATA3 to<br>AUDATA0 | E18, E17,<br>F18, D17 | I/O | AUD data            | Branch trace mode: Branch destination address output pins.              |
| debugger II<br>(AUD-II) |                       |                       |     |                     | RAM monitor mode: Monitor address input/data input/output pins.         |
|                         | AUDRST                | D16                   | I   | AUD reset           | AUD-II reset signal input pin   |
|                         | AUDMD                 | C17                   | I   | AUD mode            | Mode select signal input pin  |
|                         |                       |                       |     |                     | Branch trace mode: Input a low level                                    |
|                         |                       |                       |     |                     | RAM monitor mode: Input a high level                                    |
|                         | AUDCK                 | G18                   | I/O | AUD clock           | Branch trace mode: Clock output pin                                     |
|                         |                       |                       |     |                     | RAM monitor mode: Clock input pin                                       |
|                         | AUDSYNC               | D18                   | I/O | AUD sync<br>signal  | Branch trace mode: Data start position identification signal output pin |
|                         |                       |                       |     |                     | RAM monitor mode: Data start position identification signal input pin   |
| I/O ports               | POD                   | P4                    | I   | Port output disable | Input pin for pin driving control when a general port is set for output |
|                         | PA15 to PA0           | C18,                  | I/O | Port A              | General input/output port pins.   |
|                         |                       | B19, J18,             |     |                     | Input or output can be specified bit by bit.                            |
|                         |                       | G19,                  |     |                     |   |
|                         |                       | D20,<br>E20,          |     |                     |   |
|                         |                       | H19, F20,             |     |                     |   |
|                         |                       | C19,                  |     |                     |   |
|                         |                       | E19,                  |     |                     |   |
|                         |                       | D19,                  |     |                     |   |
|                         |                       | G17,                  |     |                     |   |
|                         |                       | F19, B20,             |     |                     |   |
|                         |                       | H18, C20              |     |                     |   |
|                         | PB14 to PB0           | G20,                  | I/O | Port B              | General input/output port pins.   |
|                         |                       | K18,                  |     |                     | Input or output can be specified bit by bit.                            |
|                         |                       | M18,                  |     |                     |   |
|                         |                       | M20,                  |     |                     |   |
|                         |                       | N20,<br>N19,          |     |                     |   |
|                         |                       | P20, J19,             |     |                     |   |
|                         |                       | L18, H20,             |     |                     |   |
|                         |                       | J20, K19,             |     |                     |   |
|                         |                       | L17, K20,             |     |                     |   |
|                         |                       | M19                   |     |                     |   |

| Classi-   |             |  |     |        |  |
|-----------|-------------|--|-----|--------|--|
| fication  | Symbol      | Pin No.                                  | I/O | Name   | Function                                     |
| I/O ports | PC15 to PC0 | N18,                                     | I/O | Port C | General input/output port pins.              |
|           |             | R20,                                     |     |        | Input or output can be specified bit by bit. |
|           |             | R18,                                     |     |        |  |
|           |             | V20,                                     |     |        |  |
|           |             | W20,                                     |     |        |  |
|           |             | V19,                                     |     |        |  |
|           |             | U18, T18,<br>P19,                        |     |        |  |
|           |             | N17, T20,                                |     |        |  |
|           |             | R19, T19,                                |     |        |  |
|           |             | P18,                                     |     |        |  |
|           |             | U20, U19                                 |     |        |  |
|           | PD13 to PD0 | B6, B7,                                  | I/O | Port D | General input/output port pins.              |
|           |             | C10, A9,                                 |     |        | Input or output can be specified bit by bit. |
|           |             | D10,                                     |     |        | pat or catput car. 20 opcomes 21t by 21th    |
|           |             | C11, A4,                                 |     |        |  |
|           |             | A5, C9,                                  |     |        |  |
|           |             | B8, A6,                                  |     |        |  |
|           |             | A7, A8,                                  |     |        |  |
|           |             | B9                                       |     |        |  |
|           | PE13 to PE0 | D4, D5,                                  | I/O | Port E | General input/output port pins.              |
|           |             | A2, A3,                                  |     |        | Input or output can be specified bit by bit. |
|           |             | C8, D8,                                  |     |        |  |
|           |             | C6, C5,                                  |     |        |  |
|           |             | C4, B3,                                  |     |        |  |
|           |             | B4, D6,                                  |     |        |  |
|           |             | B5, C7                                   |     |        |  |
|           | PF15 to PF0 | V11, Y13,                                |     | Port F | General input/output port pins.              |
|           |             | V13, Y18,                                |     |        | Input or output can be specified bit by bit. |
|           |             | V14,                                     |     |        |  |
|           |             | W15,                                     |     |        |  |
|           |             | Y19,                                     |     |        |  |
|           |             | W16,                                     |     |        |  |
|           |             | Y14,                                     |     |        |  |
|           |             | W13,                                     |     |        |  |
|           |             | V12, Y15,<br>Y16,                        |     |        |  |
|           |             | U13,                                     |     |        |  |
|           |             | W14, Y17                                 |     |        |  |
|           |             | •• |     |        |  |

| Classi-<br>fication | Symbol      | Pin No.  | I/O | Name          | Function                                     |
|---------------------|-------------|--|-----|---------------|--|
| I/O ports           | PG15 to PG0 | G1, H3,  | I/O | Port G        | General input/output port pins.              |
|                     |             | L2, L3,<br>M1, M4,<br>M2, N1,<br>H2, H1,<br>J1, J3,<br>J2, K4,<br>L1, K3 |     |               | Input or output can be specified bit by bit. |
|                     | PH5 to PH0  | B1, E3,  | I/O | Port H        | General input/output port pins.              |
|                     |             | F4, B2,<br>D3, C3  |     |               | Input or output can be specified bit by bit. |
|                     | PJ9 to PJ0  | V18, V17,  | I/O | Port J        | General input/output port pins.              |
|                     |             | V16,<br>U17, T17,<br>V15,<br>W19,<br>U15,<br>W18,<br>W17                 |     |               | Input or output can be specified bit by bit. |
|                     | PK11 to PK0 | G2, H4,  | I/O | Port K        | General input/output port pins.              |
|                     |             | D2, C2,<br>F1, E1,<br>G3, F2,<br>D1, C1,<br>E2, F3                       |     |               | Input or output can be specified bit by bit. |
|                     | PL8 to PL0  | P3, P4,  | I/O | Port L        | General input/output port pins.              |
|                     |             | N3, R2,<br>P2, R1,<br>N2, M3,<br>P1                                      |     |               | Input or output can be specified bit by bit. |
| Other               | N.C.        | Y1   | -   | No connection | No connection should be made with this pin.  |

### 1.3.3 List of Pins

Table 1.3 List of Pins

| A18       ASEMD         C14       MD4         C13       MD3         D12       MD2         A12       MD1         C12       MD0         A11       FWE         B16       MD_CLK1         D14       MD_CLK0         B15       MD_CLKP         D11       HSTBY         B12       RES         C15       NMI         A14       EXTAL         A15       XTAL     |
|--|
| C13         MD3           D12         MD2           A12         MD1           C12         MD0           A11         FWE           B16         MD_CLK1           D14         MD_CLK0           B15         MD_CLKP           D11         HSTBY           B12         RES           C15         NMI           A14         EXTAL           A15         XTAL |
| D12       MD2         A12       MD1         C12       MD0         A11       FWE         B16       MD_CLK1         D14       MD_CLK0         B15       MD_CLKP         D11       HSTBY         B12       RES         C15       NMI         A14       EXTAL         A15       XTAL   |
| A12       MD1         C12       MD0         A11       FWE         B16       MD_CLK1         D14       MD_CLK0         B15       MD_CLKP         D11       HSTBY         B12       RES         C15       NMI         A14       EXTAL         A15       XTAL   |
| C12       MD0         A11       FWE         B16       MD_CLK1         D14       MD_CLK0         B15       MD_CLKP         D11       HSTBY         B12       RES         C15       NMI         A14       EXTAL         A15       XTAL   |
| A11       FWE         B16       MD_CLK1         D14       MD_CLK0         B15       MD_CLKP         D11       HSTBY         B12       RES         C15       NMI         A14       EXTAL         A15       XTAL   |
| B16         MD_CLK1           D14         MD_CLK0           B15         MD_CLKP           D11         HSTBY           B12         RES           C15         NMI           A14         EXTAL           A15         XTAL   |
| D14         MD_CLK0           B15         MD_CLKP           D11         HSTBY           B12         RES           C15         NMI           A14         EXTAL           A15         XTAL   |
| B15         MD_CLKP           D11         HSTBY           B12         RES           C15         NMI           A14         EXTAL           A15         XTAL   |
| D11         HSTBY           B12         RES           C15         NMI           A14         EXTAL           A15         XTAL   |
| B12         RES           C15         NMI           A14         EXTAL           A15         XTAL   |
| C15 NMI A14 EXTAL A15 XTAL   |
| A14 EXTAL A15 XTAL   |
| A15 XTAL   |
|  |
|  |
| A13 CK   |
| B11 WDTOVF   |
| B17 TRST   |
| B18 TCK  |
| D15 TMS  |
| A19 TDI  |
| C16 TDO  |
| C17 AUDMD  |
| D16 AUDRST   |
| G18 AUDCK  |
| D18 AUDSYNC  |
| D17 AUDATA3  |
| F18 AUDATA2  |

| Pin Number | MCU Expansion Mode, MCU Single Chip Mode, Boot Mode,<br>User Program Mode, User Boot Mode |
|------------|---|
| E17        | AUDATA1   |
| E18        | AUDATA0   |
| C18        | PA0/A0  |
| B19        | PA1/A1  |
| C19        | PA2/A2  |
| E19        | PA3/A3  |
| D19        | PA4/A4  |
| G17        | PA5/A5  |
| F19        | PA6/A6  |
| B20        | PA7/A7  |
| H18        | PA8/A8  |
| C20        | PA9/A9  |
| J18        | PA10/A10  |
| G19        | PA11/A11  |
| D20        | PA12/A12  |
| E20        | PA13/A13  |
| H19        | PA14/A14  |
| F20        | PA15/A15  |
| G20        | PB0/A16/MOSIA   |
| K18        | PB1/A17/MISOA   |
| J19        | PB2/A18/MOSIB   |
| L18        | PB3/A19/MISOB   |
| H20        | PB4/A20/CTx_B/TIF6  |
| J20        | PB5/A21/CRx_B/TIF7  |
| K19        | PB6/WE0   |
| L17        | PB7/WE1   |
| K20        | PB8/WAIT/TOE20  |
| M19        | PB9/RD  |
| M18        | PB10/CS0  |
| M20        | PB11/CS1/TOE21  |
| N20        | PB12/CS2/RSPCKA   |
| N19        | PB13/CS3/RSPCKB   |

| Pin Number | MCU Expansion Mode, MCU Single Chip Mode, Boot Mode,<br>User Program Mode, User Boot Mode |
|------------|---|
| P20        | PB14/RD/WR  |
| N18        | PC0/D0  |
| R20        | PC1/D1  |
| P19        | PC2/D2  |
| N17        | PC3/D3  |
| T20        | PC4/D4  |
| R19        | PC5/D5  |
| T19        | PC6/D6  |
| P18        | PC7/D7  |
| U20        | PC8/D8  |
| U19        | PC9/D9  |
| R18        | PC10/D10  |
| V20        | PC11/D11  |
| W20        | PC12/D12  |
| V19        | PC13/D13  |
| U18        | PC14/D14  |
| T18        | PC15/D15  |
| B6         | PD0/TIOC00/TIOC31   |
| B7         | PD1/TIOC01/TOE20  |
| A4         | PD2/TIOC02/TOE21/TOE52  |
| A5         | PD3/TIOC03/TOE22/TOE53  |
| C9         | PD4/TIOC10/TIOC32/TOE52   |
| B8         | PD5/TIOC11/TOE23/TOE40  |
| A6         | PD6/TIOC12/TOE41  |
| A7         | PD7/TIOC13/TOE42  |
| A8         | PD8/TIOC20/TIOC33/TOE53   |
| B9         | PD9/TIOC21/TIF0B/TOE43  |
| C10        | PD10/TIOC22/TIF1B/TOE50   |
| A9         | PD11/TIOC23/TIF2B/TOE51   |
| D10        | PD12/TCLKA/TIOC41/TIJ0  |

| Pin Number | MCU Expansion Mode, MCU Single Chip Mode, Boot Mode,<br>User Program Mode, User Boot Mode |
|------------|---|
| C11        | PD13/TCLKB/TIJ1   |
| D4         | PE0/TIA00   |
| D5         | PE1/TIA01/TIOC42/TIOC40   |
| C6         | PE2/TIA02/TIOC43/TIOC30   |
| C5         | PE3/TIA03   |
| C4         | PE4/TIA04   |
| B3         | PE5/TIA05   |
| B4         | PE6/TOE00/CTx_B   |
| D6         | PE7/TOE01/CRx_B   |
| B5         | PE8/TOE02   |
| C7         | PE9/TOE03   |
| A2         | PE10/TOE10  |
| A3         | PE11/TOE11  |
| C8         | PE12/TOE12  |
| D8         | PE13/TOE13  |
| V11        | PF0/TOD00B/TIF6   |
| Y13        | PF1/TOD01B/TIF7   |
| Y14        | PF2/TOD02B/TIF8   |
| W13        | PF3/TOD03B/TIF9   |
| V12        | PF4/TOD10B/TIF10  |
| Y15        | PF5/TOD11B/TIF11  |
| Y16        | PF6/TOD12B/TIF12  |
| U13        | PF7/TOD13B/TIF13  |
| W14        | PF8/TOD20B/TIF14  |
| Y17        | PF9/TOD21B/TIF15  |
| V13        | PF10/TOD22B/TIF16   |
| Y18        | PF11/TOD23B/TIF17   |
| V14        | PF12/TOD30B/TIF18   |
| W15        | PF13/TOD31B/TIF19   |
| Y19        | PF14/TOD32B/CTx_B/TxD_A   |
| W16        | PF15/TOD33B/CRx_B/RxD_A   |

| Pin Number | MCU Expansion Mode, MCU Single Chip Mode, Boot Mode,<br>User Program Mode, User Boot Mode |
|------------|---|
| G1         | PG0/TOD00A/SSLA0  |
| H3         | PG1/TOD01A/SSLA1  |
| H2         | PG2/TOD02A/SSLA2  |
| H1         | PG3/TOD03A/SSLA3  |
| J1         | PG4/TOD10A/SSLA4/SSLB3  |
| J3         | PG5/TOD11A/SSLA5/SSLC3  |
| J2         | PG6/TOD12A/SSLB0  |
| K4         | PG7/TOD13A/SSLB1  |
| L1         | PG8/TOD20A/SSLB2/TIF6   |
| K3         | PG9/TOD21A/SSLC0/TIF7   |
| L2         | PG10/TOD22A/SSLC1/TIF8  |
| L3         | PG11/TOD23A/SSLC2/TIF9  |
| M1         | PG12/TOD30A/SSLA4/TIF10   |
| M4         | PG13/TOD31A/SSLA5/TIF11   |
| M2         | PG14/TOD32A/SSLA6/TIF12   |
| N1         | PG15/TOD33A/SSLA7/TIF13   |
| B1         | PH0/ADTRG_A/TIF0A   |
| E3         | PH1/ADTRG_B/TIF1A   |
| F4         | PH2/TIF2A   |
| B2         | PH3/TIF3  |
| D3         | PH4/TIF4  |
| C3         | PH5/TIF5  |
| V18        | PJ0/TxD_A/CTx_A/CTx_A&CTx_B   |
| V17        | PJ1/RxD_A/CRx_A/CRx_A&CRx_B   |
| V16        | PJ2/TxD_A/CTx_C/CTx_A&CTx_B&CTx_C   |
| U17        | PJ3/RxD_A/CRx_C/CRx_A&CRx_B&CRx_C   |
| T17        | PJ4/SCK_A/ADEND_B/TIJ0  |
| V15        | PJ5/TxD_A   |
| W19        | PJ6/RxD_A   |
| U15        | PJ7/SCK_B/ADEND_A/TIJ1  |
| W18        | PJ8/TxD_B   |
| W17        | PJ9/RxD_B   |

| Pin Number | MCU Expansion Mode, MCU Single Chip Mode, Boot Mode,<br>User Program Mode, User Boot Mode |
|------------|---|
| G2         | PK0/SCK_C/RSPCKA/UBCTRG   |
| H4         | PK1/TxD_C/MOSIA   |
| F1         | PK2/RxD_C/MISOA   |
| E1         | PK3/SCK_D/RSPCKB  |
| G3         | PK4/TxD_D/MOSIB   |
| F2         | PK5/RxD_D/MISOB   |
| D1         | PK6/SCK_E   |
| C1         | PK7/TxD_E   |
| E2         | PK8/RxD_E   |
| F3         | PK9/RSPCKC  |
| D2         | PK10/MOSIC  |
| C2         | PK11/MISOC  |
| P3         | PL0/IRQ0  |
| P4         | PL1/TOE20/IRQ1/POD  |
| N3         | PL2/TOE21/IRQ2  |
| R2         | PL3/TOE22/IRQ3  |
| P2         | PL4/TOE23/IRQ4  |
| R1         | PL5/TOE30/IRQ5  |
| N2         | PL6/TOE31/IRQ6  |
| M3         | PL7/TOE32/IRQ7  |
| P1         | PL8/TOE33   |
| W11        | AN_A0   |
| U11        | AN_A1   |
| Y11        | AN_A2   |
| V10        | AN_A3   |
| U10        | AN_A4   |
| Y9         | AN_A5   |
| V9         | AN_A6   |
| W9         | AN_A7   |
| U9         | AN_A8   |
| Y8         | AN_A9   |

| Pin Number | MCU Expansion Mode, MCU Single Chip Mode, Boot Mode,<br>User Program Mode, User Boot Mode |
|------------|---|
| Y7         | AN_A10  |
| Y6         | AN_A11  |
| W8         | AN_A12  |
| V8         | AN_A13  |
| Y5         | AN_A14  |
| V7         | AN_A15  |
| W7         | AN_A16  |
| W6         | AN_A17  |
| W5         | AN_A18  |
| U7         | AN_A19  |
| V6         | AN_A20  |
| U6         | AN_A21  |
| V5         | AN_A22  |
| U5         | AN_A23  |
| Y2         | AN_A24  |
| W2         | AN_A25  |
| V3         | AN_A26  |
| V4         | AN_A27  |
| Y3         | AVREFH_A  |
| W3         | AVREFL_A  |
| T3         | AN_B40  |
| U4         | AN_B41  |
| U3         | AN_B42  |
| T4         | AN_B43  |
| R3         | AN_B44  |
| V2         | AN_B45  |
| R4         | AN_B46  |
| U2         | AN_B47  |
| T2         | AN_B48  |
| W1         | AVREFH_B  |

| Pin Number | MCU Expansion Mode, MCU Single Chip Mode, Boot Mode,<br>User Program Mode, User Boot Mode |
|------------|---|
| V1         | AVREFL_B  |
| A17        | PLLV <sub>cc</sub>  |
| A16        | PLLV <sub>ss</sub>  |
| B10        | V <sub>CL</sub>   |
| K2         | $V_{\scriptscriptstyleCL}$  |
| L19        | $V_{\scriptscriptstyleCL}$  |
| W12        | V <sub>cL</sub>   |
| B14        | $V_{cc}$  |
| D7         | V <sub>cc</sub>   |
| D13        | $V_cc$  |
| F17        | V <sub>cc</sub>   |
| G4         | $V_cc$  |
| L4         | V <sub>cc</sub>   |
| M17        | V <sub>cc</sub>   |
| U8         | V <sub>cc</sub>   |
| U14        | V <sub>cc</sub>   |
| H17        | PV <sub>cc</sub> 1  |
| J17        | PV <sub>cc</sub> 1  |
| K17        | PV <sub>cc</sub> 1  |
| P17        | PV <sub>cc</sub> 1  |
| R17        | PV <sub>cc</sub> 1  |
| D9         | PV <sub>cc</sub> 2  |
| E4         | PV <sub>cc</sub> 2  |
| J4         | PV <sub>cc</sub> 2  |
| N4         | PV <sub>cc</sub> 2  |
| U12        | PV <sub>cc</sub> 2  |
| U16        | PV <sub>cc</sub> 2  |
| A1         | $V_{ss}$  |
| A10        | $V_{SS}$  |
| A20        | V <sub>ss</sub>   |
| B13        | $V_{ss}$  |

| J9  | $V_{ss}$        |
|-----|-----------------|
| J10 | V <sub>ss</sub> |
| J11 | V <sub>ss</sub> |
| J12 | V <sub>ss</sub> |
| K1  | V <sub>ss</sub> |
| K9  | $V_{ss}$        |
| K10 | $V_{ss}$        |
| K11 | $V_{ss}$        |
| K12 | $V_{ss}$        |
| L9  | $V_{ss}$        |
| L10 | $V_{ss}$        |
| L11 | $V_{ss}$        |
| L12 | $V_{ss}$        |
| L20 | $V_{ss}$        |
| M9  | V <sub>ss</sub> |
| M10 | $V_{ss}$        |
| M11 | $V_{ss}$        |
| M12 | $V_{ss}$        |
| Y12 | $V_{ss}$        |
| Y20 | $V_{ss}$        |
| T1  | $AV_cc$         |
| W10 | $AV_cc$         |
| Y4  | $AV_cc$         |
| U1  | $AV_{ss}$       |
| W4  | $AV_{ss}$       |
| Y10 | $AV_{ss}$       |
| Y1  | NC              |

SH72546R Section 2 CPU

# Section 2 CPU

### 2.1 Data Format

Figure 2.1 shows the data format supported by the SH-2A/SH2A-FPU.

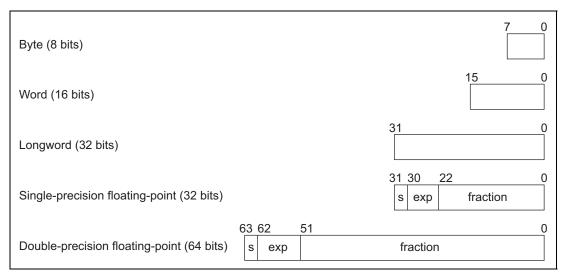


Figure 2.1 Data Format

### 2.2 Register Descriptions

The register set consists of five types of registers: sixteen 32-bit general registers, four 32-bit control registers, four 32-bit system registers, floating-point registers, and floating-point system registers.

### 2.2.1 General Registers

Figure 2.2 shows the general registers.

The general registers consists of 16 registers, numbered R0 to R15, and are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and restoring the status register (SR) and program counter (PC) in exception handling is accomplished by referencing the stack using R15.

Section 2 CPU SH72546R

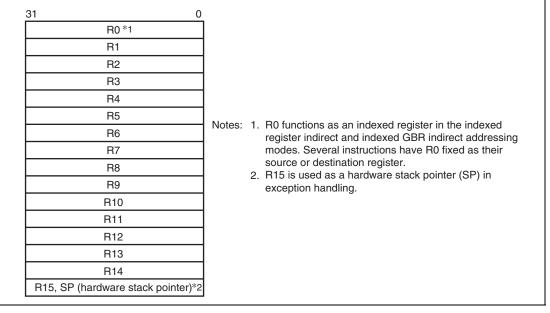


Figure 2.2 General Registers

### 2.2.2 Control Registers

The control registers consist of four 32-bit registers: the status register (SR), the global base register (GBR), the vector base register (VBR), and the jump table base register (TBR).

The status register indicates instruction processing states.

The global base register functions as a base address for the GBR indirect addressing mode to transfer data to the registers of on-chip peripheral modules.

The vector base register functions as the base address of the exception handling vector area (including interrupts).

The jump table base register functions as the base address of the function table area.

SH72546R Section 2 CPU

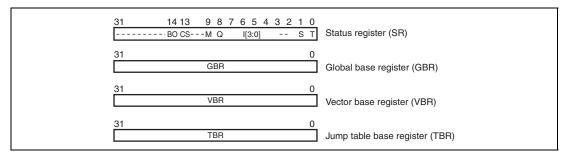


Figure 2.3 Control Registers

### (1) Status Register (SR)

| Bit:           | 31 | 30  | 29  | 28 | 27 | 26 | 25  | 24  | 23  | 22  | 21  | 20  | 19 | 18 | 17  | 16  |
|----------------|----|-----|-----|----|----|----|-----|-----|-----|-----|-----|-----|----|----|-----|-----|
|                | -  | -   | -   | -  | -  | -  | -   | -   | -   | -   | -   | -   | -  | -  | -   | -   |
| Initial value: | 0  | 0   | 0   | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0  | 0  | 0   | 0   |
| R/W:           | R  | R   | R   | R  | R  | R  | R   | R   | R   | R   | R   | R   | R  | R  | R   | R   |
|                |    |     |     |    |    |    |     |     |     |     |     |     |    |    |     |     |
| Bit:           | 15 | 14  | 13  | 12 | 11 | 10 | 9   | 8   | 7   | 6   | 5   | 4   | 3  | 2  | 1   | 0   |
|                | 1  | во  | CS  | -  | -  | -  | М   | Q   |     | 1[3 | :0] |     | -  | -  | S   | Т   |
| Initial value: | 0  | 0   | 0   | 0  | 0  | 0  | -   | -   | 1   | 1   | 1   | 1   | 0  | 0  | -   | -   |
| R/W:           | R  | R/W | R/W | R  | R  | R  | R/W | R/W | R/W | R/W | R/W | R/W | R  | R  | R/W | R/W |

| Bit      | Bit Name | Initial<br>Value | R/W | Description   |
|----------|----------|------------------|-----|---|
| 31 to 15 | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 14       | ВО       | 0                | R/W | BO Bit  |
|          |          |                  |     | Indicates the register bank has overflowed.   |
| 13       | CS       | 0                | R/W | CS Bit  |
|          |          |                  |     | Indicates, in CLIP instruction execution, the value has exceeded the saturation upper-limit value or fallen below the saturation lower-limit value. |
| 12 to 10 | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.  |

SH72546R Section 2 CPU

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 9      | М        | _                | R/W | M Bit  |
| 8      | Q        | _                | R/W | Q Bit  |
|        |          |                  |     | Used by the DIV0S, DIV0U, and DIV1 instructions.                     |
| 7 to 4 | I[3:0]   | 1111             | R/W | Interrupt Mask Level   |
| 3, 2   | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 1      | S        | _                | R/W | S Bit  |
|        |          |                  |     | Specifies a saturation operation for a MAC instruction.              |
| 0      | Т        | _                | R/W | T Bit  |
|        |          |                  |     | True/false condition or carry/borrow bit                             |

#### **(2)** Global Base Register (GBR)

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

#### **Vector Base Register (VBR) (3)**

VBR is referenced as the branch destination base address when an exception or an interrupt occurs.

#### **(4) Jump Table Base Register (TBR)**

TBR is referenced as the start address of a function table located in memory in a JSR/N@@(disp8,TBR) table-referencing subroutine call instruction.

#### 2.2.3 **System Registers**

The system registers consist of four 32-bit registers: the high and low multiply and accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). MACH and MACL store the results of multiply or multiply and accumulate operations. PR stores the return address from a subroutine procedure. PC indicates the program address being executed and controls the flow of the processing.

SH72546R Section 2 CPU

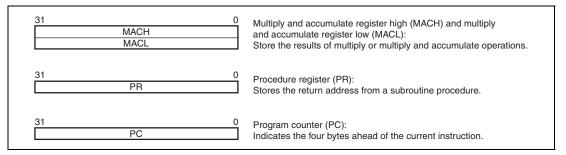


Figure 2.4 System Registers

# (1) Multiply and Accumulate Register High (MACH) and Multiply and Accumulate Register Low (MACL)

MACH and MACL are used as the addition value in a MAC instruction, and store the result of a MAC or MUL instruction.

### (2) Procedure Register (PR)

PR stores the return address of a subroutine call using a BSR, BSRF, or JSR instruction, and is referenced by a subroutine return instruction (RTS).

### (3) Program Counter (PC)

PC indicates the address four bytes ahead of the instruction being currently executed.

### 2.2.4 Floating-Point Registers

Figure 2.5 shows the floating-point registers. There are sixteen 32-bit floating-point registers, FPR0 to FPR15. These sixteen registers are referenced as FR0 to FR15, DR0, DR2, DR4, DR6, DR8, DR10, DR12, and DR14. The correspondence between FPRn and the referenced name is determined by the PR and SZ bits in FPSCR (see figure 2.5).

### (1) Floating-Point Registers (FPRn: 16 registers)

FPR0, FPR1, FPR2, FPR3, FPR4, FPR5, FPR6, FPR7, FPR8, FPR9, FPR10, FPR11, FPR12, FPR13, FPR14, and FPR15

# (2) Single-Precision Floating-Point Registers (FRi: 16 registers)

FR0 to FR15 are allocated to FPR0 to FPR15.

Section 2 CPU SH72546R

# (3) Double-Precision Floating-Point Registers or Single-Precision Floating-Point Register Pairs (DRi: 8 registers)

A DR register is composed of two FR registers.

 $DR0 = \{FPR0, FPR1\}, DR2 = \{FPR2, FPR3\}, DR4 = \{FPR4, FPR5\}, DR6 = \{FPR4, FPR5\}, DR8 = \{FPR8, FPR9\}, DR10 = \{FPR10, FPR11\}, DR12 = \{FPR12, FPR13\}, and DR14 = \{FPR14, FPR15\}$ 

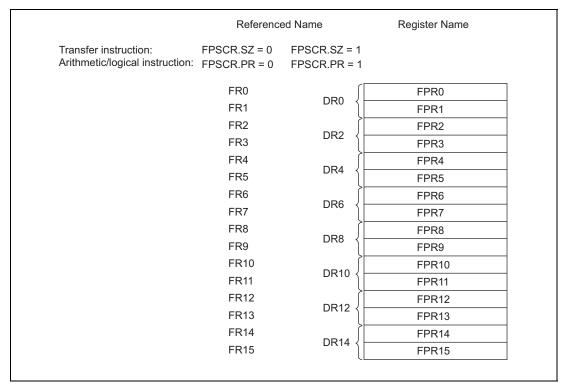


Figure 2.5 Floating-Point Registers

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

### 2.2.5 Floating-Point System Registers

### (1) Floating-Point Communication Register (FPUL)

Data is transferred between an FPU register and a CPU register via FPUL.

SH72546R Section 2 CPU

# (2) Floating Point Status/Control Register (FPSCR)

| Bit:                   | 31       | 30       | 29       | 28       | 27     | 26       | 25       | 24       | 23       | 22       | 21       | 20       | 19       | 18       | 17       | 16       |
|------------------------|----------|----------|----------|----------|--------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                        | -        | -        | -        | -        | -      | -        | -        | -        | -        | QIS      | -        | SZ       | PR       | DN       | Ca       | use      |
| Initial value:<br>R/W: | 0<br>R   | 0<br>R   | 0<br>R   | 0<br>R   | 0<br>R | 0<br>R   | 0<br>R   | 0<br>R   | 0<br>R   | 0<br>R/W | 0<br>R   | 0<br>R/W | 0<br>R/W | 1<br>R/W | 0<br>R/W | 0<br>R/W |
| Bit:                   | 15       | 14       | 13       | 12       | 11     | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|                        | Cause    |          |          |          |        | Enable   |          |          |          |          | Flag     |          |          | RM[      | 1:0]     |          |
| Initial value:         | 0<br>B/M | 0<br>R/W | 0<br>R/W | 0<br>B/W | 0      | 0<br>B/W | 0<br>R/W | 0<br>B/W | 0<br>B/W | 0<br>R/M | 0<br>R/W | 0<br>B/M | 0<br>B/W | 0        | 0<br>B/W | 1        |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 31 to 23 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 22       | QIS      | 0                | R/W | sNaN is treated as qNaN or $\pm \infty$ . Valid only when the V bit in the FPU exception enable field (Enable) is set to 1.                |
|          |          |                  |     | 0: Processed as qNaN or $\pm \infty$   |
|          |          |                  |     | 1: Exception generated (processed same as sNaN)  |
| 21       | _        | 0                | R   | Reserved   |
|          |          |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 20       | SZ       | 0                | R/W | Transfer Size Mode   |
|          |          |                  |     | 0: Sets the size of an FMOV instruction to 32 bits.  |
|          |          |                  |     | 1: Sets the size of an FMOV instruction to 32-bit pair (64 bits).  |
| 19       | PR       | 0                | R/W | Precision Mode   |
|          |          |                  |     | <ol><li>Executes floating-point instructions in single precision.</li></ol>  |
|          |          |                  |     | Executes floating-point instructions in double precision (the result of an instruction with no support for double-precision is undefined). |

Section 2 CPU SH72546R

| Bit      | Bit Name | Initial<br>Value | R/W | Description   |
|----------|----------|------------------|-----|---|
| 18       | DN       | 1                | R/W | Denormalization Mode  |
|          |          |                  |     | This bit is always set to 1.  |
|          |          |                  |     | 1: A denormalized number is treated as zero.  |
| 17 to 12 | Cause    | All 0            | R/W | FPU exception cause field   |
| 11 to 7  | Enable   | All 0            | R/W | FPU exception enable field  |
| 6 to 2   | Flag     | All 0            | R/W | FPU exception flag field  |
|          |          |                  |     | When an FPU operation instruction is first executed, the FPU exception cause field is set to 0; when an FPU exception next occurs, the corresponding bit in the FPU exception cause field and FPU exception flag field is set to 1. |
|          |          |                  |     | The FPU exception flag field retains the status of an exception generated after that field was last cleared.  |
|          |          |                  |     | For bit allocation for each field, see table 2.1.   |
| 1, 0     | RM[1:0]  | 01               | R/W | Round Mode  |
|          |          |                  |     | 00: Round to nearest  |
|          |          |                  |     | 01: Round to zero   |
|          |          |                  |     | 10: Reserved  |
|          |          |                  |     | 11: Reserved  |

Table 2.1 Bit Allocation for FPU Exception Handling

|        |                                  | FPU Error<br>(E) | Invalid<br>Operation<br>(V) | Division<br>by 0 (Z) | Overflow<br>(O) | Underflow<br>(U) | Incorrect<br>(I) |
|--------|----------------------------------|------------------|-----------------------------|----------------------|-----------------|------------------|------------------|
| Cause  | FPU<br>exception<br>cause field  | Bit 17           | Bit 16                      | Bit 15               | Bit 14          | Bit 13           | Bit 12           |
| Enable | FPU<br>exception<br>enable field | None             | Bit 11                      | Bit 10               | Bit 9           | Bit 8            | Bit 7            |
| Flag   | FPU<br>exception flag<br>field   | None             | Bit 6                       | Bit 5                | Bit 4           | Bit 3            | Bit 2            |

Note: In the SH-2A, no FPU errors occur.

## 2.2.6 Register Bank

Using a register bank, high-speed register saving and restoration can be achieved for the 19 32-bit registers: general registers R0 to R14, control register GBR, and system registers MACH, MACL, and PR. The register contents are automatically saved in the bank after the CPU accepts an interrupt that uses the bank. Restoration from the bank is executed by a RESBANK instruction issued in an interrupt processing routine.

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

## 2.2.7 Initial Values of Registers

Table 2.2 lists the values of the registers after a reset.

**Table 2.2** Initial Values of Registers

| Register       | Initial Value  |
|----------------|--|
| R0 to R14      | Undefined  |
| R15 (SP)       | Value of the stack pointer in the vector address table                                     |
| SR             | Bits I[3:0] are 1111 (H'F), BO and CS are 0, reserved bits are 0, and others are undefined |
| GBR, TBR       | Undefined  |
| VBR            | H'00000000   |
| MACH, MACL, PR | Undefined  |
| PC             | Value of the program counter in the vector address table                                   |
| FPR0 to FPR15  | Undefined  |
| FPUL           | Undefined  |
| FPSCR          | H'00040001   |
|                | R0 to R14 R15 (SP) SR GBR, TBR VBR MACH, MACL, PR PC FPR0 to FPR15 FPUL                    |

### 2.3 Data Formats

### 2.3.1 Data Format in Registers

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

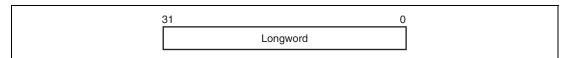


Figure 2.6 Data Format in Registers

### 2.3.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in 8-bit bytes, 16-bit words, or 32-bit longwords. A memory operand of fewer than 32 bits is stored in a register in sign-extended or zero-extended form.

A word operand should be accessed at a word boundary (an even address of multiple of two bytes: address 2n), and a longword operand at a longword boundary (an even address of multiple of four bytes: address 4n). Otherwise, an address error will occur. A byte operand can be accessed at any address.

Only big-endian byte order can be selected for the data format.

Data formats in memory are shown in figure 2.7.

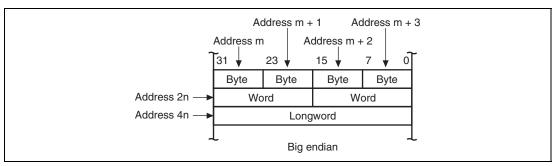


Figure 2.7 Data Formats in Memory

#### 2.3.3 Immediate Data Format

Byte (8-bit) immediate data is located in an instruction code. Immediate data accessed by the MOV, ADD, and CMP/EQ instructions is sign-extended and handled in registers as longword data. Immediate data accessed by the TST, AND, OR, and XOR instructions is zero-extended and handled as longword data. Consequently, AND instructions with immediate data always clear the upper 24 bits of the destination register.

20-bit immediate data is located in the code of a MOVI20 or MOVI20S 32-bit transfer instruction. The MOVI20 instruction stores immediate data in the destination register in sign-extended form. The MOVI20S instruction shifts immediate data by eight bits in the upper direction, and stores it in the destination register in sign-extended form.

Word or longword immediate data is not located in the instruction code, but rather is stored in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

See examples given in section 2.4.1 (10), Immediate Data.

### 2.4 Instruction Features

### 2.4.1 RISC-Type Instruction Set

The CPU has a RISC-type instruction set, which features following functions.

### (1) 16-Bit Fixed-Length Instructions

Basic instructions have a fixed length of 16 bits, improving program code efficiency.

## (2) 32-Bit Fixed-Length Instructions

The SH-2A/SH2A-FPU additionally features 32-bit fixed-length instructions, improving performance and ease of use.

### (3) One Instruction per Cycle

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

### (4) Data Length

The standard data length for all operations is a longword. Memory can be accessed in bytes, words, or longwords. Byte or word data in memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It is also handled as longword data.

Table 2.3 Sign Extension of Word Data

| SH2-A/SH2A-FPU CPU |                        | Description  | Exampl | e of Other CPU |
|--------------------|------------------------|--|--------|----------------|
| MOV.W<br>ADD       | @(disp,PC),R1<br>R1,R0 | Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction. | ADD.W  | #H'1234,R0     |
|                    | • • •                  |  |        |                |
| .DATA.W            | Н'1234                 |  |        |                |

Note: @(disp, PC) accesses the immediate data.

## (5) Load-Store Architecture

Basic operations are executed between registers. For operations that involve memory access, data is loaded to the registers and executed (load-store architecture). Instructions such as AND that manipulate bits, however, are executed directly in memory.

### (6) Delayed Branch Instructions

With the exception of some instructions, unconditional branch instructions, etc., are executed as delayed branch instructions. With a delayed branch instruction, the branch is taken after execution of the instruction immediately following the delayed branch instruction. This reduces disturbance of the pipeline control when a branch is taken.

In a delayed branch, the actual branch operation occurs after execution of the slot instruction. However, instruction execution such as register updating excluding the actual branch operation, is performed in the order of delayed branch instruction  $\rightarrow$  delay slot instruction. For example, even though the contents of the register holding the branch destination address are changed in the delay slot, the branch destination address remains as the register contents prior to the change.

**Table 2.4 Delayed Branch Instructions** 

| SH2-A/SH | 12A-FPU CPU | Description                          | Example | of Other CPU |
|----------|-------------|--------------------------------------|---------|--------------|
| BRA      | TRGET       | Executes the ADD before branching to | ADD.W   | R1,R0        |
| ADD      | R1,R0       | TRGET.                               | BRA     | TRGET        |

### (7) Unconditional Branch Instructions with No Delay Slot

The SH-2A/SH2A-FPU additionally features unconditional branch instructions in which a delay slot instruction is not executed. This eliminates unnecessary NOP instructions, and so reduces the code size.

## (8) Multiply/Multiply-and-Accumulate Operations

16-bit  $\times$  16-bit  $\rightarrow$  32-bit multiply operations are executed in one to two cycles. 16-bit  $\times$  16-bit + 64-bit  $\rightarrow$  64-bit multiply-and-accumulate operations are executed in two to three cycles. 32-bit  $\times$  32-bit  $\rightarrow$  64-bit multiply and 32-bit  $\times$  32-bit  $\rightarrow$  64-bit multiply-and-accumulate operations are executed in two to four cycles.

## (9) T Bit

The T bit in the status register (SR) changes according to the result of the comparison. Whether a conditional branch is taken or not taken depends upon the T bit condition (true/false). The number of instructions that change the T bit is kept to a minimum to improve the processing speed.

Table 2.5 T Bit

| SH2-A/SH2A-FPU CPU |        | Description Example of Other                    |       | e of Other CPU |
|--------------------|--------|---|-------|----------------|
| CMP/GE             | R1,R0  | T bit is set when $R0 \ge R1$ .                 | CMP.W | R1,R0          |
| BT                 | TRGET0 | The program branches to TRGET0                  | BGE   | TRGET0         |
| BF                 | TRGET1 | when $R0 \ge R1$ and to TRGET1 when $R0 < R1$ . | BLT   | TRGET1         |
| ADD                | #-1,R0 | T bit is not changed by ADD.                    | SUB.W | #1,R0          |
| CMP/EQ             | #0,R0  | T bit is set when $R0 = 0$ .                    | BEQ   | TRGET          |
| BT                 | TRGET  | The program branches if $R0 = 0$ .              |       |                |

#### (10) Immediate Data

Byte immediate data is located in an instruction code. Word or longword immediate data is not located in instruction codes but in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

With the SH-2A/SH2A-FPU, 17- to 28-bit immediate data can be located in an instruction code. However, for 21- to 28-bit immediate data, an OR instruction must be executed after the data is transferred to a register.

**Table 2.6** Immediate Data Accessing

| Classification   | SH-2A/SH2A-FPU CPU |               | Example of Other CPU |                |  |
|------------------|--------------------|---------------|----------------------|----------------|--|
| 8-bit immediate  | MOV                | #H'12,R0      | MOV.B                | #H'12,R0       |  |
| 16-bit immediate | MOVI20             | #H'1234,R0    | MOV.W                | #H'1234,R0     |  |
| 20-bit immediate | MOVI20             | #H'12345,R0   | MOV.L                | #H'12345,R0    |  |
| 28-bit immediate | MOVI20S            | #H'12345,R0   | MOV.L                | #H'1234567,R0  |  |
|                  | OR                 | #H'67,R0      |                      |                |  |
| 32-bit immediate | MOV.L              | @(disp,PC),R0 | MOV.L                | #H'12345678,R0 |  |
|                  |                    |               |                      |                |  |
|                  | .DATA.L            | Н'12345678    |                      |                |  |

Note: @(disp, PC) accesses the immediate data.

#### (11) Absolute Address

When data is accessed by an absolute address, the absolute address value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in register indirect addressing mode.

With the SH-2A/SH2A-FPU, when data is referenced using an absolute address not exceeding 28 bits, it is also possible to transfer immediate data located in the instruction code to a register and to reference the data in register indirect addressing mode. However, when referencing data using an absolute address of 21 to 28 bits, an OR instruction must be used after the data is transferred to a register.

Table 2.7 Absolute Address Accessing

| Classification  | SH-2A/SH2A-FPU CPU |               | Example of Other CPU |                |
|-----------------|--------------------|---------------|----------------------|----------------|
| Up to 20 bits   | MOVI20             | #H'12345,R1   | MOV.B                | @H'12345,R0    |
|                 | MOV.B              | @R1,R0        |                      |                |
| 21 to 28 bits   | MOVI20S            | #H'12345,R1   | MOV.B                | @H'1234567,R0  |
|                 | OR                 | #H'67,R1      |                      |                |
|                 | MOV.B              | @R1,R0        |                      |                |
| 29 bits or more | MOV.L              | @(disp,PC),R1 | MOV.B                | @H'12345678,R0 |
|                 | MOV.B              | @R1,R0        |                      |                |
|                 |                    |               |                      |                |
|                 | .DATA.L            | H'12345678    |                      |                |

## (12) 16-Bit/32-Bit Displacement

When data is accessed by 16-bit or 32-bit displacement, the displacement value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in the indexed indirect register addressing mode.

Table 2.8 Displacement Accessing

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

## 2.4.2 Addressing Modes

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

Table 2.9 Addressing Modes and Effective Addresses

| Addressing<br>Mode   | Instruction Format   | Effective Address Calculation   | Equation                          |
|--|--|---|-----------------------------------|
| Register direct  | Rn   | The effective address is register Rn. (The operand is the contents of register Rn.) | _                                 |
| Register indirect  | @Rn  | The effective address is the contents of register Rn.                               | Rn                                |
|  |  | Rn Rn   |                                   |
| Register indirect  | @Rn+   | The effective address is the contents of register                                   | Rn                                |
| with post-<br>increment  | with post- increment  A constant is added to the contents of Rn after the instruction is executed. 1 is added for a byte | (After instruction execution)   |                                   |
| operation, 2 for a word operation, and 4 for longword operation.  Rn Rn Rn Rn Rn | longword operation.  | Byte: $Rn + 1 \rightarrow Rn$   |                                   |
|  |  |   | Word: $Rn + 2 \rightarrow Rn$     |
|  |  | 1/2/4   | Longword: $Rn + 4 \rightarrow Rn$ |

| Addressing<br>Mode                  | Instruction<br>Format | Effective Address Calculation   | Equation  |
|-------------------------------------|-----------------------|---|---|
| Register indirect with predecrement | @-Rn                  | The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation.  | Byte:<br>$Rn - 1 \rightarrow Rn$<br>Word:<br>$Rn - 2 \rightarrow Rn$                                      |
|                                     |                       | Rn - 1/2/4 - Rn - 1/2/4   | Longword:<br>$Rn - 4 \rightarrow Rn$<br>(Instruction is<br>executed with<br>Rn after this<br>calculation) |
| Register indirect with displacement | @(disp:4,<br>Rn)      | The effective address is the sum of Rn and a 4-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation.  Rn  disp (zero-extended)  Rn + disp × 1/2/4 | Byte:<br>Rn + disp<br>Word:<br>Rn + disp × 2<br>Longword:<br>Rn + disp × 4                                |
| Register indirect with displacement | @(disp:12<br>,Rn)     | The effective address is the sum of Rn and a 12-bit displacement (disp). The value of disp is zero-extended.  Rn  Rn + disp  (zero-extended)  | Byte:<br>Rn + disp<br>Word:<br>Rn + disp<br>Longword:<br>Rn + disp  |
| Indexed register indirect           | @(R0,Rn)              | The effective address is the sum of Rn and R0.  Rn  Rn + R0   | Rn + R0   |

| Addressing<br>Mode                       | Instruction<br>Format  | Effective Address Calculation   | Equation                             |
|--|------------------------|---|--------------------------------------|
| GBR indirect with displacement           | @(disp:8,<br>GBR)      | an 8-bit displacement (disp). The value of disp is  | Byte:<br>GBR + disp                  |
|  |                        | zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation.     | Word:<br>GBR + disp ×<br>2           |
|  |                        | GBR  disp (zero-extended)  + disp × 1/2/4   | Longword:<br>GBR + disp ×<br>4       |
|  |                        | 1/2/4   |                                      |
| Indexed GBR indirect                     | @(R0,<br>GBR)          | The effective address is the sum of GBR value and R0.   | GBR + R0                             |
|  |                        | GBR + R0  |                                      |
|  |                        | RO  |                                      |
| TBR duplicate indirect with displacement | @@<br>(disp:8,<br>TBR) | The effective address is the sum of TBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and is multiplied by 4. | Contents of address (TBR + disp × 4) |
|  |                        | disp (zero-extended) + disp × 4 (TBR  |                                      |
|  |                        | 4 + disp × 4)   |                                      |

| Addressing<br>Mode            | Instruction<br>Format | Effective Address Calculation  | Equation  |
|-------------------------------|-----------------------|--|---|
| PC indirect with displacement | @(disp:8,<br>PC)      | The effective address is the sum of PC value and an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked.  PC  (for longword)  PC + disp × 2 or PC & H'FFFFFFFC  disp (zero-extended) | Word: PC + disp × 2 Longword: PC & H'FFFFFFC + disp × 4 |
| PC relative                   | disp:8                | The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 8-bit displacement (disp).  PC  disp (sign-extended)  PC + disp × 2  | PC + disp × 2   |
|                               | disp:12               | The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp).  PC  disp (sign-extended)  PC + disp × 2   | PC + disp × 2   |

| Addressing<br>Mode | Instruction<br>Format | Effective Address Calculation   | Equation |
|--------------------|-----------------------|---|----------|
| PC relative        | Rn                    | The effective address is the sum of PC value and Rn.  PC PC+Rn Rn   | PC + Rn  |
| Immediate          | #imm:20               | The 20-bit immediate data (imm) for the MOVI20 instruction is sign-extended.  31  | _        |
|                    |                       | The 20-bit immediate data (imm) for the MOVI20S instruction is shifted by eight bits to the left, the upper bits are sign-extended, and the lower bits are padded with zero.  31 27 8 0     imm (20 bits) | _        |
|                    | #imm:8                | The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions is zero-extended.   | _        |
|                    | #imm:8                | The 8-bit immediate data (imm) for the MOV, ADD and CMP/EQ instructions is sign-extended.   | , —      |
|                    | #imm:8                | The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and then quadrupled.  | _        |
|                    | #imm:3                | The 3-bit immediate data (imm) for the BAND, BOR, BXOR, BST, BLD, BSET, and BCLR instructions indicates the target bit location.  | _        |

#### 2.4.3 Instruction Format

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

• xxxx: Instruction code

• mmmm: Source register

• nnnn: Destination register

• iiii: Immediate data

• dddd: Displacement

**Table 2.10 Instruction Formats** 

| Instruction Formats                 | Source Operand                              | Destination<br>Operand                        | Example          |
|-------------------------------------|---|---|------------------|
| 0 format                            | _   | _   | NOP              |
| 15                                  |   |   |                  |
| n format                            | _   | nnnn: Register direct                         | MOVT Rn          |
| 15 0<br>  XXXX   nnnn   XXXX   XXXX | Control register or system register         | nnnn: Register direct                         | STS MACH, Rn     |
|                                     | R0 (Register direct)                        | nnnn: Register direct                         | DIVU R0,Rn       |
|                                     | Control register or system register         | nnnn: Register indirect with pre-decrement    | STC.L SR,@-Rn    |
|                                     | mmmm: Register direct                       | R15 (Register indirect with pre-decrement)    | MOVMU.L Rm,@-R15 |
|                                     | R15 (Register indirect with post-increment) | nnnn: Register direct                         | MOVMU.L @R15+,Rn |
|                                     | R0 (Register direct)                        | nnnn: (Register indirect with post-increment) | MOV.L R0,@Rn+    |

| Instruction Formats                  | Source Operand   | Destination<br>Operand                     | Example                |
|--------------------------------------|--|--|------------------------|
| m format                             | mmmm: Register direct  | Control register or system register        | LDC Rm, SR             |
| xxxx mmmm xxxx xxxx                  | mmm: Register indirect with post-increment                             | Control register or system register        | LDC.L @Rm+,SR          |
|                                      | mmmm: Register indirect  | _  | JMP @Rm                |
|                                      | mmmm: Register indirect with predecrement                              | R0 (Register direct)                       | MOV.L @-Rm,R0          |
|                                      | mmmm: PC relative using Rm   | _  | BRAF Rm                |
| nm format                            | mmmm: Register direct  | nnnn: Register direct                      | ADD Rm,Rn              |
| 15 0                                 | mmmm: Register direct  | nnnn: Register indirect                    | MOV.L Rm,@Rn           |
| SOST MAIN MAINE SOST                 | mmm: Register indirect with post-increment (multiply-and-accumulate)   | MACH, MACL                                 | MAC.W @Rm+,@Rn+        |
|                                      | nnnn*: Register indirect with post-increment (multiply-and-accumulate) |  |                        |
|                                      | mmmm: Register indirect with post-increment                            | nnnn: Register direct                      | MOV.L @Rm+,Rn          |
|                                      | mmmm: Register direct  | nnnn: Register indirect with pre-decrement | MOV.L Rm,@-Rn          |
|                                      | mmmm: Register direct  | nnnn: Indexed register indirect            | MOV.L<br>Rm,@(R0,Rn)   |
| md format  15 0  xxxx xxxx mmmm dddd | mmmmdddd: Register indirect with displacement                          | R0 (Register direct)                       | MOV.B<br>@(disp,Rm),R0 |

| Instruction Formats                    | Source Operand   | Destination<br>Operand                              | Example                             |
|--|--|---|-------------------------------------|
| nd4 format  15 0  xxxx xxxx nnnn dddd  | R0 (Register direct)                                     | nnnndddd:<br>Register indirect with<br>displacement | MOV.B<br>R0,@(disp,Rn)              |
| nmd format  15 0  xxxx nnnn mmmm dddd  | mmmm: Register direct                                    | nnnndddd: Register indirect with displacement       | MOV.L<br>Rm,@(disp,Rn)              |
|  | mmmmdddd: Register indirect with displacement            | nnnn: Register direct                               | MOV.L<br>@(disp,Rm),Rn              |
| nmd12 format  32                       | mmmm: Register direct                                    | nnnndddd: Register indirect with displacement       | MOV.L<br>Rm,@(disp12,Rn)            |
| 15 0 xxxx dddd dddd dddd               | mmmmdddd: Register indirect with displacement            | nnnn: Register direct                               | MOV.L<br>@(disp12,Rm),Rn            |
| d format  15 0  xxxx xxxx dddd dddd    | dddddddd: GBR<br>indirect with<br>displacement           | R0 (Register direct)                                | MOV.L<br>@(disp,GBR),R0             |
|  | R0 (Register direct)                                     | ddddddd: GBR<br>indirect with<br>displacement       | MOV.L<br>R0,@(disp,GBR)             |
|  | dddddddd: PC relative with displacement                  | R0 (Register direct)                                | MOVA<br>@(disp,PC),R0               |
|  | dddddddd: TBR<br>duplicate indirect with<br>displacement | _   | JSR/N<br>@@(disp8,TBR)              |
|  | dddddddd: PC relative                                    | _   | BF label                            |
| d12 format  15 0  xxxx dddd dddd dddd  | ddddddddddd: PC relative                                 | _   | BRA label<br>(label = disp +<br>PC) |
| nd8 format  15 0  xxxxx nnnn dddd dddd | dddddddd: PC relative with displacement                  | nnnn: Register direct                               | MOV.L<br>@(disp,PC),Rn              |

| Source Operand  | Destination<br>Operand                               | Example   |
|---|--|---|
| iiiiiiiii:<br>Immediate   | Indexed GBR indirect                                 | AND.B<br>#imm,@(R0,GBR)   |
| iiiiiiii:<br>Immediate  | R0 (Register direct)                                 | AND #imm,R0   |
| iiiiiiii:<br>Immediate  | _  | TRAPA #imm  |
| iiiiiiii:<br>Immediate  | nnnn: Register direct                                | ADD #imm,Rn   |
| _   | _  | BLD #imm3,Rn  |
|   | nnnn: Register direct                                | BST #imm3,Rn  |
| iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii  | nnnn: Register direct                                | MOVI20<br>#imm20, Rn  |
|   |  |   |
| nnnnddddddd<br>ddd: Register<br>indirect with<br>displacement<br>iii: Immediate | _  | BLD.B<br>#imm3,@(disp12,Rn<br>)   |
| _   | nnnnddddddddddd: Register indirect with displacement | BST.B<br>#imm3,@(disp12,Rn<br>)   |
|   | iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii               | Source Operand  iiiiiiii: Indexed GBR indirect Immediate  iiiiiiii: R0 (Register direct)  Immediate  iiiiiiii: — Immediate  iiiiiiii: Immediate  nnnn: Register direct iii: Immediate  nnnnddddddddd dddd: Register indirect with displacement iii: Immediate  nnnndddddddddddddddddddddddddddddddd |

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

# 2.5 Instruction Set

## 2.5.1 Instruction Set by Classification

Table 2.11 lists the instructions according to their classification.

**Table 2.11 Classification of Instructions** 

| Classification | Types | Operation Code | Function  | No. of<br>Instructions |
|----------------|-------|----------------|---|------------------------|
| Data transfer  | 13    | MOV            | Data transfer                                   | 62                     |
|                |       |                | Immediate data transfer                         |                        |
|                |       |                | Peripheral module data transfer                 |                        |
|                |       |                | Structure data transfer                         |                        |
|                |       |                | Reverse stack transfer                          |                        |
|                |       | MOVA           | Effective address transfer                      |                        |
|                |       | MOVI20         |   |                        |
|                |       | MOVI20S        | 20-bit immediate data transfer                  |                        |
|                |       |                | 8-bit left-shit                                 |                        |
|                |       | MOVML          | R0-Rn register save/restore                     |                        |
|                |       | MOVMU          | Rn-R14 and PR register save/restore             | <del></del>            |
|                |       | MOVRT          | T bit inversion and transfer to Rn              |                        |
|                |       | MOVT           | T bit transfer                                  |                        |
|                |       | MOVU           | Unsigned data transfer                          |                        |
|                |       | NOTT           | T bit inversion                                 |                        |
|                |       | PREF           | Prefetch to operand cache                       | <del></del>            |
|                |       | SWAP           | Swap of upper and lower bytes                   | <del></del>            |
|                |       | XTRCT          | Extraction of the middle of registers connected |                        |

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

| Classification | Types | Operation<br>Code | Function                          | No. of<br>Instructions |
|----------------|-------|-------------------|-----------------------------------|------------------------|
| Logic          | 6     | AND               | Logical AND                       | 14                     |
| operations     |       | NOT               | Bit inversion                     | <del></del>            |
|                |       | OR                | Logical OR                        | <del></del>            |
|                |       | TAS               | Memory test and bit set           | <del></del>            |
|                |       | TST               | Logical AND and T bit set         | <del></del>            |
|                |       | XOR               | Exclusive OR                      | <del></del>            |
| Shift          | 12    | ROTL              | One-bit left rotation             | 16                     |
|                |       | ROTR              | One-bit right rotation            | <del></del>            |
|                |       | ROTCL             | One-bit left rotation with T bit  | <del></del>            |
|                |       | ROTCR             | One-bit right rotation with T bit | <del></del>            |
|                |       | SHAD              | Dynamic arithmetic shift          | <del></del>            |
|                |       | SHAL              | One-bit arithmetic left shift     | <del></del>            |
|                |       | SHAR              | One-bit arithmetic right shift    | <del></del>            |
|                |       | SHLD              | Dynamic logical shift             | <del></del>            |
|                |       | SHLL              | One-bit logical left shift        | <del></del>            |
|                |       | SHLLn             | n-bit logical left shift          | <u> </u>               |
|                |       | SHLR              | One-bit logical right shift       | <u> </u>               |
|                |       | SHLRn             | n-bit logical right shift         | <u> </u>               |

| Classification | Types | Operation<br>Code | Function   | No. of<br>Instructions |
|----------------|-------|-------------------|--|------------------------|
| Branch         | 10    | BF                | Conditional branch, conditional delayed branch (branch when T = 0) | 15                     |
|                |       | ВТ                | Conditional branch, conditional delayed branch (branch when T = 1) | _                      |
|                |       | BRA               | Unconditional delayed branch                                       | <del>-</del>           |
|                |       | BRAF              | Unconditional delayed branch                                       | <del>-</del>           |
|                |       | BSR               | Delayed branch to subroutine procedure                             | _                      |
|                |       | BSRF              | Delayed branch to subroutine procedure                             | _                      |
|                |       | JMP               | Unconditional delayed branch                                       | <del>-</del>           |
|                |       | JSR               | Branch to subroutine procedure                                     |                        |
|                |       |                   | Delayed branch to subroutine procedure                             |                        |
|                |       | RTS               | Return from subroutine procedure                                   | _                      |
|                |       |                   | Delayed return from subroutine procedure                           |                        |
|                |       | RTV/N             | Return from subroutine procedure with Rm $\rightarrow$ R0 transfer | _                      |
| System         | 14    | CLRT              | T bit clear  | 36                     |
| control        |       | CLRMAC            | MAC register clear   | _                      |
|                |       | LDBANK            | Register restoration from specified register bank entry            | _                      |
|                |       | LDC               | Load to control register   | _                      |
|                |       | LDS               | Load to system register  |                        |
|                |       | NOP               | No operation   | _                      |
|                |       | RESBANK           | Register restoration from register bank                            | <del>_</del>           |
|                |       | RTE               | Return from exception handling                                     | _                      |
|                |       | SETT              | T bit set  | <del>_</del>           |
|                |       | SLEEP             | Transition to power-down mode                                      | <del>_</del>           |
|                |       | STBANK            | Register save to specified register bank entry                     | _                      |
|                |       | STC               | Store control register data  | _                      |
|                |       | STS               | Store system register data   | _                      |
|                |       | TRAPA             | Trap exception handling  | _                      |

|                |       | Operation           |  | No. of       |
|----------------|-------|---------------------|--|--------------|
| Classification | Types | Code                | Function   | Instructions |
| Floating-point | 19    | FABS                | Floating-point absolute value                          | 48           |
| instructions   |       | FADD                | Floating-point addition                                | =            |
|                |       | FCMP                | Floating-point comparison                              | •            |
|                |       | FCNVDS              | Conversion from double-precision to single-precision   | -            |
|                |       | FCNVSD              | Conversion from single-precision to double - precision | -            |
|                |       | FDIV                | Floating-point division                                | -            |
|                |       | FLDI0               | Floating-point load immediate 0                        | -            |
|                |       | FLDI1               | Floating-point load immediate 1                        | -            |
|                |       | FLDS                | Floating-point load into system register FPUL          | -            |
|                |       | FLOAT               | Conversion from integer to floating-point              | -            |
|                |       | FMAC                | Floating-point multiply and accumulate operation       | -            |
|                |       | FMOV                | Floating-point data transfer                           | -            |
|                |       | FMUL Floating-point | Floating-point multiplication                          | -            |
|                |       | FNEG                | Floating-point sign inversion                          | -            |
|                |       | FSCHG               | SZ bit inversion                                       | -            |
|                |       | FSQRT               | Floating-point square root                             | -            |
|                |       | FSTS                | Floating-point store from system register FPUL         | -            |
|                |       | FSUB                | Floating-point subtraction                             | -            |
|                |       | FTRC                | Floating-point conversion with rounding to integer     | <del>-</del> |

| Classification      | Types | Operation Code | Function                                  | No. of<br>Instructions |
|---------------------|-------|----------------|---|------------------------|
| FPU-related         | 2     | LDS            | Load into floating-point system register  | 8                      |
| CPU instructions    |       | STS            | Store from floating-point system register | <del></del>            |
| Bit<br>manipulation | 10    | BAND           | Bit AND                                   | 14                     |
|                     |       | BCLR           | Bit clear                                 |                        |
|                     |       | BLD            | Bit load                                  |                        |
|                     |       | BOR            | Bit OR                                    |                        |
|                     |       | BSET           | Bit set                                   |                        |
|                     |       | BST            | Bit store                                 |                        |
|                     |       | BXOR           | Bit exclusive OR                          |                        |
|                     |       | BANDNOT        | Bit NOT AND                               |                        |
|                     |       | BORNOT         | Bit NOT OR                                |                        |
|                     |       | BLDNOT         | Bit NOT load                              |                        |
| Total:              | 112   |                |   | 253                    |

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

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

- Notes: 1. Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states will be increased in cases such as the following:
  - a. When there is a conflict between an instruction fetch and a data access
  - b. When the destination register of a load instruction (memory  $\rightarrow$  register) is the same as the register used by the next instruction.
  - 2. Depending on the operand size, displacement is scaled by  $\times 1$ ,  $\times 2$ , or  $\times 4$ . For details, refer to the SH-2A, SH2A-FPU Software Manual.

## 2.5.2 Data Transfer Instructions

**Table 2.12 Data Transfer Instructions** 

|             |                |                  |  |                          |       | Co   | ompatib | ility                  |
|-------------|----------------|------------------|--|--------------------------|-------|------|---------|------------------------|
| Instruction | on             | Instruction Code | Operation  | Execu-<br>tion<br>Cycles | T Bit | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| MOV         | #imm, Rn       | 1110nnnniiiiiiii | $\operatorname{imm} \to \operatorname{sign}$ extension $\to \operatorname{Rn}$ | 1                        | _     | Yes  | Yes     |                        |
| MOV.W       | @(disp, PC),Rn | 1001nnnndddddddd | $(disp \times 2 + PC) \rightarrow sign$<br>extension $\rightarrow Rn$          | 1                        | _     | Yes  | Yes     |                        |
| MOV.L       | @(disp, PC),Rn | 1101nnnndddddddd | $(disp \times 4 + PC) \rightarrow Rn$  | 1                        | _     | Yes  | Yes     |                        |
| MOV         | Rm, Rn         | 0110nnnnmmmm0011 | $Rm \rightarrow Rn$  | 1                        | _     | Yes  | Yes     |                        |
| MOV.B       | Rm, @Rn        | 0010nnnnmmmm0000 | $Rm \rightarrow (Rn)$  | 1                        | _     | Yes  | Yes     |                        |
| MOV.W       | Rm, @Rn        | 0010nnnnmmmm0001 | $Rm \rightarrow (Rn)$  | 1                        | _     | Yes  | Yes     |                        |
| MOV.L       | Rm, @Rn        | 0010nnnnmmmm0010 | $Rm \rightarrow (Rn)$  | 1                        | _     | Yes  | Yes     |                        |
| MOV.B       | @Rm, Rn        | 0110nnnnmmm0000  | $(Rm) \rightarrow sign extension$<br>$\rightarrow Rn$                          | 1                        | _     | Yes  | Yes     |                        |
| MOV.W       | @Rm, Rn        | 0110nnnnmmmm0001 | $(Rm) \rightarrow sign extension$<br>$\rightarrow Rn$                          | 1                        | _     | Yes  | Yes     |                        |
| MOV.L       | @Rm, Rn        | 0110nnnnmmmm0010 | $(Rm) \rightarrow Rn$  | 1                        |       | Yes  | Yes     |                        |
| MOV.B       | Rm, @-Rn       | 0010nnnnmmmm0100 | $Rn-1 \rightarrow Rn, Rm \rightarrow (Rn)$                                     | 1                        | _     | Yes  | Yes     |                        |
| MOV.W       | Rm, @-Rn       | 0010nnnnmmmm0101 | $Rn-2 \rightarrow Rn, Rm \rightarrow (Rn)$                                     | 1                        | _     | Yes  | Yes     |                        |
| MOV.L       | Rm, @-Rn       | 0010nnnnmmmm0110 | $Rn-4 \rightarrow Rn, Rm \rightarrow (Rn)$                                     | 1                        | _     | Yes  | Yes     |                        |
| MOV.B       | @Rm+, Rn       | 0110nnnnmmm0100  | (Rm) → sign extension → Rn, Rm + 1 → Rm  | 1                        | _     | Yes  | Yes     |                        |
| MOV.W       | @Rm+, Rn       | 0110nnnnmmm0101  | $(Rm) \rightarrow sign extension$<br>$\rightarrow Rn, Rm + 2 \rightarrow Rm$   | 1                        | _     | Yes  | Yes     |                        |
| MOV.L       | @Rm+, Rn       | 0110nnnnmmm0110  | $(Rm) \rightarrow Rn, Rm + 4 \rightarrow Rm$                                   | 1                        | _     | Yes  | Yes     |                        |
| MOV.B       | R0, @(disp,Rn) | 10000000nnnndddd | $R0 \rightarrow (disp + Rn)$   | 1                        |       | Yes  | Yes     |                        |
| MOV.W       | R0, @(disp,Rn) | 10000001nnnndddd | $R0 \rightarrow (disp \times 2 + Rn)$  | 1                        |       | Yes  | Yes     |                        |
| MOV.L       | Rm, @(disp,Rn) | 0001nnnnmmmmdddd | $Rm \rightarrow (disp \times 4 + Rn)$  | 1                        |       | Yes  | Yes     |                        |
| MOV.B       | @(disp, Rm),R0 | 10000100mmmmdddd | $(disp + Rm) \rightarrow sign$<br>extension $\rightarrow R0$                   | 1                        |       | Yes  | Yes     |                        |

| MOV.L @ MOV.B Rn MOV.B @ MOV.W @ MOV.L @ MOV.L MOV.B MOV.B MOV.B RO MOV.B RO MOV.B RO MOV.B RO MOV.B @ | (disp, Rm),R0<br>(disp, Rm),Rn<br>m,@(R0,Rn)<br>m,@(R0,Rn) | Instruction Code  10000101mmmmdddd   | Operation $(\text{disp} \times 2 + \text{Rm}) \rightarrow \\ \text{sign extension} \rightarrow \text{R0}$ | Execu-<br>tion<br>Cycles | T Bit | SH2E | 0114 | SH-2A/<br>SH2A- |
|--|--|--------------------------------------|---|--------------------------|-------|------|------|-----------------|
| MOV.L @ MOV.B Rn MOV.B @ MOV.W @ MOV.L @ MOV.L MOV.B MOV.B MOV.B RO MOV.B RO MOV.B RO MOV.B RO MOV.B @ | (disp, Rm),Rn<br>m,@(R0,Rn)                                |                                      | ` '   |                          |       | J    | SH4  | FPU             |
| MOV.B Rn MOV.W Rn MOV.L Rn MOV.B @ I MOV.W @ I MOV.L @ I MOV.B R0 MOV.W R0 MOV.L R0 MOV.B @ I  | m,@(R0,Rn)   | 0101nnnnmmmmdddd                     | *   | 1                        | _     | Yes  | Yes  |                 |
| MOV.W Rn MOV.L Rn MOV.B @ I MOV.W @ I MOV.L @ I MOV.B R0 MOV.W R0 MOV.L R0 MOV.L R0 MOV.B @ I  |  |                                      | $(disp \times 4 + Rm) \rightarrow Rn$   | 1                        | _     | Yes  | Yes  |                 |
| MOV.L Rn MOV.W @ 1 MOV.L @ 1 MOV.B R0 MOV.W R0 MOV.L R0 MOV.L R0 MOV.B @ 1   | m @(B0 Bn)   | 0000nnnnmmm0100                      | $Rm \rightarrow (R0 + Rn)$  | 1                        | _     | Yes  | Yes  |                 |
| MOV.B @ (MOV.L @ (MOV.B RO) MOV.W RO) MOV.L RO) MOV.B @ (MOV.B @ (MOV.B @ (MOV.B )   | , 0 (0,)   | 0000nnnnmmmm0101                     | $Rm \rightarrow (R0 + Rn)$  | 1                        | _     | Yes  | Yes  |                 |
| MOV.W @ OME MOV.B RO MOV.W RO MOV.L RO MOV.B @ OME MOV.B   | m,@(R0,Rn)   | 0000nnnnmmmm0110                     | $Rm \rightarrow (R0 + Rn)$  | 1                        | _     | Yes  | Yes  |                 |
| MOV.L @ (MOV.B RO) MOV.W RO) MOV.L RO) MOV.B @ (   | (R0,Rm),Rn   | 0000nnnnmmm1100                      | $(R0 + Rm) \rightarrow$<br>sign extension $\rightarrow Rn$  | 1                        | _     | Yes  | Yes  |                 |
| MOV.B RO<br>MOV.W RO<br>MOV.L RO<br>MOV.B @ (  | (R0,Rm),Rn   | 0000nnnnmmm1101                      | $(R0 + Rm) \rightarrow$<br>sign extension $\rightarrow Rn$  | 1                        | _     | Yes  | Yes  |                 |
| MOV.W RO<br>MOV.L RO<br>MOV.B @  | (R0,Rm),Rn   | 0000nnnnmmmm1110                     | $(R0 + Rm) \rightarrow Rn$  | 1                        | _     | Yes  | Yes  |                 |
| MOV.B @  | 0,@(disp,GBR)  | 11000000dddddddd                     | R0 → (disp + GBR)   | 1                        | _     | Yes  | Yes  |                 |
| MOV.B @  | 0,@(disp,GBR)  | 11000001dddddddd                     | $R0 \rightarrow (disp \times 2 + GBR)$  | 1                        | _     | Yes  | Yes  |                 |
|  | 0,@(disp,GBR)  | 11000010dddddddd                     | $R0 \rightarrow (disp \times 4 + GBR)$  | 1                        | _     | Yes  | Yes  |                 |
| MOV.W @  | (disp,GBR),R0  | 11000100dddddddd                     | $(disp + GBR) \rightarrow$<br>sign extension $\rightarrow R0$   | 1                        | _     | Yes  | Yes  |                 |
|  | (disp,GBR),R0  | 11000101dddddddd                     | $(disp \times 2 + GBR) \rightarrow$<br>sign extension $\rightarrow R0$                                    | 1                        | _     | Yes  | Yes  |                 |
| MOV.L @  | (disp,GBR),R0  | 11000110dddddddd                     | $(disp \times 4 + GBR) \rightarrow R0$  | 1                        | _     | Yes  | Yes  |                 |
| MOV.B R0   | 0,@Rn+   | 0100nnnn10001011                     | $R0 \rightarrow (Rn), Rn + 1 \rightarrow Rn$  | 1                        | _     |      |      | Yes             |
| MOV.W R0   | 0,@Rn+   | 0100nnnn10011011                     | $R0 \rightarrow (Rn), Rn + 2 \rightarrow$<br>Rn   | 1                        | _     |      |      | Yes             |
| MOV.L RO   | 0,@Rn+   | 0100nnnn10101011                     | $R0 \rightarrow Rn)$ , $Rn + 4 \rightarrow Rn$  | 1                        | _     |      |      | Yes             |
| MOV.B @-   | -Rm,R0   | 0100mmmm11001011                     | $Rm-1 \rightarrow Rm, (Rm) \rightarrow$ sign extension $\rightarrow R0$                                   | 1                        | _     |      |      | Yes             |
| MOV.W @-   | -Rm,R0   | 0100mmmm11011011                     | $Rm-2 \rightarrow Rm, (Rm) \rightarrow$ sign extension $\rightarrow R0$                                   | 1                        | _     |      |      | Yes             |
| MOV.L @-   | -Rm,R0   | 0100mmmm11101011                     | $Rm-4 \rightarrow Rm, (Rm) \rightarrow R0$  | 1                        | _     |      |      | Yes             |
| MOV.B Rn   | m,@(disp12,Rn)   | 0011nnnnmmmm0001<br>0000dddddddddddd | $Rm \rightarrow (disp + Rn)$  | 1                        | _     |      |      | Yes             |
| MOV.W Rn   | m,@(disp12,Rn)   | 0011nnnnmmmm0001                     | $Rm \rightarrow (disp \times 2 + Rn)$   | 1                        | _     |      |      | Yes             |

|            |                   |   |   |                          |   | Co   | ompatib | ility                  |
|------------|-------------------|---|---|--------------------------|---|------|---------|------------------------|
| Instructio | n                 | Instruction Code                        | Operation   | Execu-<br>tion<br>Cycles |   | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| MOV.L      | Rm,@(disp12,Rn)   | 0011nnnnmmmm0001                        | $Rm \rightarrow (disp \times 4 + Rn)$   | 1                        | _ |      |         | Yes                    |
|            |                   | 0010dddddddddddd                        |   |                          |   |      |         |                        |
| MOV.B      | @(disp12, Rm), Rn | 0011nnnnmmmm0001                        | $(disp + Rm) \to$   | 1                        | _ |      |         | Yes                    |
|            |                   | 0100dddddddddddd                        | sign extension → Rn   |                          |   |      |         |                        |
| MOV.W      | @(disp12, Rm), Rn | 0011nnnnmmmm0001                        | $(disp \times 2 + Rm) \to$  | 1                        | _ |      |         | Yes                    |
|            |                   | 0101dddddddddddd                        | sign extension $\rightarrow$ Rn   |                          |   |      |         |                        |
| MOV.L      | @(disp12, Rm), Rn | 0011nnnnmmmm0001                        | $(disp \times 4 + Rm) \rightarrow Rn$   | 1                        | _ |      |         | Yes                    |
|            |                   | 0110dddddddddddd                        |   |                          |   |      |         |                        |
| MOVA       | @(disp,PC),R0     | 11000111dddddddd                        | $disp \times 4 + PC \to R0$   | 1                        | _ | Yes  | Yes     |                        |
| MOVI20     | #imm20, Rn        | 0000nnnniiii0000                        | imm → sign extension  | 1                        | _ |      |         | Yes                    |
|            |                   | iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii | $\rightarrow$ Rn  |                          |   |      |         |                        |
| MOVI20S #  | #imm20, Rn        | 0000nnnniiii0001                        | imm << 8 → sign   | 1                        | _ |      |         | Yes                    |
|            |                   | iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii | $extension \to Rn$  |                          |   |      |         |                        |
| MOVML.L    | Rm, @-R15         | 0100mmmm11110001                        | R15-4 $\rightarrow$ R15, Rm $\rightarrow$<br>(R15)<br>R15-4 $\rightarrow$ R15, Rm-1 $\rightarrow$<br>(R15)<br>:<br>R15-4 $\rightarrow$ R15, R0 $\rightarrow$<br>(R15) | 1 to 16                  | _ |      |         | Yes                    |
|            |                   |   | Note: When Rm = R15,<br>read Rm as PR   |                          |   |      |         |                        |
| MOVML.L    | @R15+, Rn         | 0100nnnn11110101                        | $(R15) \to R0, R15 + 4 \to R15$<br>$(R15) \to R1, R15 + 4 \to R15$<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:<br>:                                   | 1 to 16                  | _ |      |         | Yes                    |

|             |                   |                                      |  |                          |                          | C    | ompatib | oility                 |
|-------------|-------------------|--------------------------------------|--|--------------------------|--------------------------|------|---------|------------------------|
| Instruction | ı                 | Instruction Code                     | Operation  | Execu-<br>tion<br>Cycles | T Bit                    | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| MOVMU.L     | Rm, @-R15         | 0100mmm11110000                      | R15-4 $\rightarrow$ R15, PR $\rightarrow$ (R15)<br>R15-4 $\rightarrow$ R15, R14 $\rightarrow$ (R15)<br>:<br>R15-4 $\rightarrow$ R15, Rm $\rightarrow$ (R15)                  | 1 to 16                  | _                        |      |         | Yes                    |
|             |                   |                                      | Note: When Rm = R15,<br>read Rm as PR  |                          |                          |      |         |                        |
| MOVMU.L     | @R15+, Rn         | 0100nnnn11110100                     | $(R15) \rightarrow Rn, R15 + 4 \rightarrow$<br>R15<br>$(R15) \rightarrow Rn + 1, R15 +$<br>$4 \rightarrow R15$<br>:<br>$(R15) \rightarrow R14, R15 + 4$<br>$\rightarrow R15$ | 1 to 16                  | _                        |      |         | Yes                    |
|             |                   |                                      | $(R15) \rightarrow PR$   |                          |                          |      |         |                        |
|             |                   |                                      | Note: When Rn = R15,<br>read Rm as PR  |                          |                          |      |         |                        |
| MOVRT       | Rn                | 0000nnnn00111001                     | $\sim$ T → Rn  | 1                        | _                        |      |         | Yes                    |
| MOVT        | Rn                | 0000nnnn00101001                     | $T \rightarrow Rn$   | 1                        | _                        | Yes  | Yes     |                        |
| MOVU.B      | @(disp12, Rm), Rn | 0011nnnnmmmm0001<br>1000dddddddddddd | $(disp + Rm) \rightarrow$<br>zero extension $\rightarrow Rn$   | 1                        | _                        |      |         | Yes                    |
| MOVU.W      | @(disp12, Rm), Rn | 0011nnnnmmmm0001<br>1001dddddddddddd | $(disp \times 2 + Rm) \rightarrow$<br>zero extension $\rightarrow Rn$  | 1                        | _                        |      |         | Yes                    |
| NOTT        |                   | 000000001101000                      | $\sim T \rightarrow T$   | 1                        | Ope-<br>ration<br>result |      |         | Yes                    |
| PREF        | @Rn               | 0000nnnn10000011                     | (Rn) → operand cache   | 1                        | _                        |      | Yes     |                        |
| SWAP.B      | Rm, Rn            | 0110nnnnmmm1000                      | Rm → swap lower 2<br>bytes → Rn  | 1                        | _                        | Yes  | Yes     |                        |
| SWAP.W      | Rm, Rn            | 0110nnnnmmm1001                      | $Rm \rightarrow swap upper and$ lower words $\rightarrow Rn$   | 1                        | _                        | Yes  | Yes     |                        |
| XTRCT       | Rm, Rn            | 0010nnnnmmm1101                      | Middle 32 bits of Rm:Rn → Rn   | 1                        | _                        | Yes  | Yes     |                        |

## 2.5.3 Arithmetic Operation Instructions

**Table 2.13 Arithmetic Operation Instructions** 

|            |          |                  |  |                          |                           | C    | ompatib | oility                 |
|------------|----------|------------------|--|--------------------------|---------------------------|------|---------|------------------------|
| Instructio | n        | Instruction Code | Operation  | Execu-<br>tion<br>Cycles |                           | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| ADD        | Rm, Rn   | 0011nnnnmmmm1100 | $Rn + Rm \rightarrow Rn$   | 1                        | _                         | Yes  | Yes     |                        |
| ADD        | #imm, Rn | 0111nnnniiiiiiii | $Rn + imm \rightarrow Rn$  | 1                        | _                         | Yes  | Yes     |                        |
| ADDC       | Rm, Rn   | 0011nnnnmmm1110  | $Rn + Rm + T \rightarrow Rn,$ $carry \rightarrow T$                              | 1                        | Carry                     | Yes  | Yes     |                        |
| ADDV       | Rm, Rn   | 0011nnnnmmm1111  | $Rn + Rm \rightarrow Rn,$ overflow $\rightarrow T$                               | 1                        | Over-<br>flow             | Yes  | Yes     |                        |
| CMP/EQ     | #imm, R0 | 10001000iiiiiiii | When R0 = imm, $1 \rightarrow T$<br>Otherwise, $0 \rightarrow T$                 | 1                        | Com-<br>parison<br>result | Yes  | Yes     |                        |
| CMP/EQ     | Rm, Rn   | 0011nnnnmmmm0000 | When Rn = Rm, 1 $\rightarrow$ T<br>Otherwise, 0 $\rightarrow$ T                  | 1                        | Com-<br>parison<br>result | Yes  | Yes     |                        |
| CMP/HS     | Rm,Rn    | 0011nnnnmmmm0010 | When Rn $\geq$ Rm (unsigned),<br>1 $\rightarrow$ T Otherwise, 0 $\rightarrow$ T  | 1                        | Com-<br>parison<br>result | Yes  | Yes     |                        |
| CMP/GE     | Rm, Rn   | 0011nnnnmmmm0011 | When Rn $\geq$ Rm (signed),<br>1 $\rightarrow$ T<br>Otherwise, 0 $\rightarrow$ T | 1                        | Com-<br>parison<br>result | Yes  | Yes     |                        |
| CMP/HI     | Rm, Rn   | 0011nnnnmmmm0110 | When Rn > Rm (unsigned), $1 \rightarrow T$ Otherwise, $0 \rightarrow T$          | 1                        | Com-<br>parison<br>result | Yes  | Yes     |                        |
| CMP/GT     | Rm,Rn    | 0011nnnnmmm0111  | When Rn > Rm (signed),<br>$1 \rightarrow T$<br>Otherwise, $0 \rightarrow T$      | 1                        | Com-<br>parison<br>result | Yes  | Yes     |                        |
| CMP/PL     | Rn       | 0100nnnn00010101 | When Rn > 0, 1 $\rightarrow$ T<br>Otherwise, 0 $\rightarrow$ T                   | 1                        | Com-<br>parison<br>result | Yes  | Yes     |                        |
| CMP/PZ     | Rn       | 0100nnnn00010001 | When Rn $\geq$ 0, 1 $\rightarrow$ T<br>Otherwise, 0 $\rightarrow$ T              | 1                        | Com-<br>parison<br>result | Yes  | Yes     |                        |

|           |        |                  |  |                          |                            | C    | ompatik | oility                 |
|-----------|--------|------------------|--|--------------------------|----------------------------|------|---------|------------------------|
| CLIPU.B F | n      | Instruction Code | Operation  | Execu-<br>tion<br>Cycles | T Bit                      | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| CMP/STR   | Rm, Rn | 0010nnnnmmmm1100 | When any bytes are equal, 1 → T Otherwise, 0 → T   | 1                        | Com-<br>parison<br>result  | Yes  | Yes     |                        |
| CLIPS.B   | Rn     | 0100nnnn10010001 | When Rn > $ (H'0000007F), \\ (H'0000007F) \rightarrow Rn, 1 \\ \rightarrow CS \\ when Rn < \\ (H'FFFFFF80), \\ (H'FFFFF80) \rightarrow Rn, 1 \\ \rightarrow CS $ | 1                        |                            |      |         | Yes                    |
| CLIPS.W   | Rn     | 0100nnnn10010101 | When Rn > (H'00007FFF), (H'00007FFF) $\rightarrow$ Rn, 1 $\rightarrow$ CS When Rn < (H'FFFF8000), (H'FFFF8000) $\rightarrow$ Rn, 1 $\rightarrow$ CS              | 1                        |                            |      |         | Yes                    |
| CLIPU.B   | Rn     | 0100nnnn10000001 | When Rn > (H'000000FF), (H'000000FF) → Rn, 1 → CS  | 1                        | _                          |      |         | Yes                    |
| CLIPU.W   | Rn     | 0100nnnn10000101 | When Rn > (H'0000FFFF), (H'0000FFFF) $\rightarrow$ Rn, 1 $\rightarrow$ CS  | 1                        | _                          |      |         | Yes                    |
| DIV1      | Rm, Rn | 0011nnnnmmmm0100 | 1-step division (Rn ÷<br>Rm)   | 1                        | Calcu-<br>lation<br>result | Yes  | Yes     |                        |
| DIVOS     | Rm, Rn | 0010nnnnmmmm0111 | $\begin{split} & \text{MSB of Rn} \to Q, \\ & \text{MSB of Rm} \to M,  \text{M} \land Q \\ & \to T \end{split}$  | 1                        | Calcu-<br>lation<br>result | Yes  | Yes     |                        |
| DIV0U     |        | 000000000011001  | $0 \rightarrow M/Q/T$  | 1                        | 0                          | Yes  | Yes     |                        |
| DIVS      | R0, Rn | 0100nnnn10010100 | Signed operation of Rn $\div$ R0 $\rightarrow$ Rn 32 $\div$ 32 $\rightarrow$ 32 bits   | 36                       | _                          |      |         | Yes                    |

|            |            |                  |  |                          |                           | Co   | ompatib | ility                  |
|------------|------------|------------------|--|--------------------------|---------------------------|------|---------|------------------------|
| Instructio | n          | Instruction Code | Operation  | Execu-<br>tion<br>Cycles | T Bit                     | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| DIVU       | R0, Rn     | 0100nnnn10000100 | Unsigned operation of Rn $\div$ R0 $\rightarrow$ Rn 32 $\div$ 32 $\rightarrow$ 32 bits                   | 34                       | _                         |      |         | Yes                    |
| DMULS.L    | Rm, Rn     | 0011nnnnmmmm1101 | Signed operation of Rn $\times$ Rm $\rightarrow$ MACH, MACL $32 \times 32 \rightarrow 64$ bits           | 2                        | _                         | Yes  | Yes     |                        |
| DMULU.L    | Rm, Rn     | 0011nnnnmmmm0101 | Unsigned operation of Rn $\times$ Rm $\rightarrow$ MACH, MACL $32 \times 32 \rightarrow 64$ bits         | 2                        | _                         | Yes  | Yes     |                        |
| DT         | Rn         | 0100nnnn00010000 | $Rn - 1 \rightarrow Rn$<br>When Rn is 0, 1 $\rightarrow$ T<br>When Rn is not 0, 0 $\rightarrow$ T        | 1                        | Com-<br>parison<br>result | Yes  | Yes     |                        |
| EXTS.B     | Rm, Rn     | 0110nnnnmmm1110  | Byte in Rm is sign-extended $\rightarrow$ Rn   | 1                        | _                         | Yes  | Yes     |                        |
| EXTS.W     | Rm, Rn     | 0110nnnnmmm1111  | Word in Rm is sign-extended → Rn   | 1                        | _                         | Yes  | Yes     |                        |
| EXTU.B     | Rm, Rn     | 0110nnnnmmm1100  | Byte in Rm is zero-extended → Rn   | 1                        | _                         | Yes  | Yes     |                        |
| EXTU.W     | Rm, Rn     | 0110nnnnmmm1101  | Word in Rm is zero-extended → Rn   | 1                        | _                         | Yes  | Yes     |                        |
| MAC.L      | @Rm+, @Rn+ | 0000nnnnmmm1111  | Signed operation of (Rn) $\times$ (Rm) + MAC $\rightarrow$ MAC $32 \times 32 + 64 \rightarrow 64$ bits   | 4                        | _                         | Yes  | Yes     |                        |
| MAC.W      | @Rm+, @Rn+ | 0100nnnmmmm1111  | Signed operation of (Rn) $\times$ (Rm) + MAC $\rightarrow$ MAC 16 $\times$ 16 + 64 $\rightarrow$ 64 bits | 3                        | _                         | Yes  | Yes     |                        |
| MUL.L      | Rm, Rn     | 0000nnnnmmm0111  | $Rn \times Rm \rightarrow MACL$ $32 \times 32 \rightarrow 32 \text{ bits}$                               | 2                        | _                         | Yes  | Yes     |                        |
| MULR       | R0, Rn     | 0100nnnn10000000 | $R0 \times Rn \rightarrow Rn$<br>$32 \times 32 \rightarrow 32$ bits                                      | 2                        |                           |      |         | Yes                    |
| MULS.W     | Rm, Rn     | 0010nnnnmmmm1111 | Signed operation of Rn $\times$ Rm $\rightarrow$ MACL 16 $\times$ 16 $\rightarrow$ 32 bits               | 1                        |                           | Yes  | Yes     |                        |

|            |        |                  |  |                          |               | Co   | mpatib | ility                  |
|------------|--------|------------------|--|--------------------------|---------------|------|--------|------------------------|
| Instructio | n      | Instruction Code | Operation  | Execu-<br>tion<br>Cycles |               | SH2E | SH4    | SH-2A/<br>SH2A-<br>FPU |
| MULU.W     | Rm, Rn | 0010nnnnmmmm1110 | Unsigned operation of Rn $\times$ Rm $\rightarrow$ MACL 16 $\times$ 16 $\rightarrow$ 32 bits         | 1                        | _             | Yes  | Yes    |                        |
| NEG        | Rm, Rn | 0110nnnnmmmm1011 | $0\text{-Rm} \to \text{Rn}$  | 1                        | _             | Yes  | Yes    |                        |
| NEGC       | Rm, Rn | 0110nnnnmmm1010  | $0-Rm-T \rightarrow Rn$ , borrow $\rightarrow T$   | 1                        | Borrow        | Yes  | Yes    |                        |
| SUB        | Rm, Rn | 0011nnnnmmmm1000 | $Rn-Rm \rightarrow Rn$   | 1                        | _             | Yes  | Yes    |                        |
| SUBC       | Rm, Rn | 0011nnnnmmmm1010 | Rn-Rm-T $\rightarrow$ Rn, borrow $\rightarrow$ T   | 1                        | Borrow        | Yes  | Yes    |                        |
| SUBV       | Rm, Rn | 0011nnnnmmmm1011 | $\begin{array}{l} \text{Rn-Rm} \rightarrow \text{Rn, underflow} \\ \rightarrow \text{T} \end{array}$ | 1                        | Over-<br>flow | Yes  | Yes    |                        |

## 2.5.4 Logic Operation Instructions

**Table 2.14 Logic Operation Instructions** 

|            |                  |   |  |                          |                | C    | ompatik | ility                  |
|------------|------------------|---|--|--------------------------|----------------|------|---------|------------------------|
| Instructio | on               | Instruction Code                        | Operation  | Execu-<br>tion<br>Cycles |                | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| AND        | Rm, Rn           | 0010nnnnmmm1001                         | $Rn \& Rm \rightarrow Rn$  | 1                        | _              | Yes  | Yes     |                        |
| AND        | #imm, R0         | 11001001iiiiiiii                        | R0 & imm → R0  | 1                        | _              | Yes  | Yes     |                        |
| AND.B      | #imm, @(R0, GBR) | 11001101iiiiiiii                        | $ (R0 + GBR) \& imm \rightarrow $ $ (R0 + GBR) $   | 3                        | _              | Yes  | Yes     |                        |
| NOT        | Rm, Rn           | 0110nnnnmmmm0111                        | $\sim$ Rm → Rn   | 1                        | _              | Yes  | Yes     |                        |
| OR         | Rm, Rn           | 0010nnnnmmm1011                         | $Rn \mid Rm \rightarrow Rn$  | 1                        | _              | Yes  | Yes     |                        |
| OR         | #imm, R0         | 11001011iiiiiiii                        | R0   imm $\rightarrow$ R0  | 1                        |                | Yes  | Yes     |                        |
| OR.B       | #imm, @(R0, GBR) | 110011111111111111111111111111111111111 | $ (R0 + GBR) \mid imm \rightarrow $ $ (R0 + GBR) $   | 3                        | _              | Yes  | Yes     |                        |
| TAS.B      | @Rn              | 0100nnnn00011011                        | When (Rn) is 0, 1 $\rightarrow$ T<br>Otherwise, 0 $\rightarrow$ T,<br>1 $\rightarrow$ MSB of(Rn) | 3                        | Test<br>result | Yes  | Yes     |                        |

|            |                  |                  |   |                          |                | Co   | ompatib | ility                  |
|------------|------------------|------------------|---|--------------------------|----------------|------|---------|------------------------|
| Instructio | n                | Instruction Code | Operation   | Execu-<br>tion<br>Cycles | T Bit          | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| TST        | Rm, Rn           | 0010nnnnmmm1000  | Rn & Rm When the result is 0, 1 $\rightarrow T$ Otherwise, $0 \rightarrow T$          | 1                        | Test<br>result | Yes  | Yes     |                        |
| TST        | #imm, R0         | 11001000iiiiiii  | R0 & imm  When the result is 0, 1 $\rightarrow$ T  Otherwise, 0 $\rightarrow$ T       | 1                        | Test<br>result | Yes  | Yes     |                        |
| TST.B      | #imm, @(R0, GBR) | 11001100iiiiiii  | (R0 + GBR) & imm When the result is 0, 1 $\rightarrow$ T Otherwise, 0 $\rightarrow$ T | 3                        | Test<br>result | Yes  | Yes     |                        |
| XOR        | Rm, Rn           | 0010nnnnmmmm1010 | $Rn \wedge Rm \rightarrow Rn$   | 1                        | _              | Yes  | Yes     |                        |
| XOR        | #imm, R0         | 11001010iiiiiiii | R0 ^ imm $\rightarrow$ R0   | 1                        | _              | Yes  | Yes     |                        |
| XOR.B      | #imm, @(R0, GBR) | 11001110iiiiiiii | $(R0 + GBR) \land imm \rightarrow$<br>(R0 + GBR)                                      | 3                        | _              | Yes  | Yes     |                        |

## 2.5.5 Shift Instructions

## **Table 2.15 Shift Instructions**

|             |        |                  |   |                          |     | C    | ompatib | ility                  |
|-------------|--------|------------------|---|--------------------------|-----|------|---------|------------------------|
| Instruction | on     | Instruction Code | Operation   | Execu-<br>tion<br>Cycles |     | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| ROTL        | Rn     | 0100nnnn00000100 | $T \leftarrow Rn \leftarrow MSB$  | 1                        | MSB | Yes  | Yes     |                        |
| ROTR        | Rn     | 0100nnnn00000101 | $LSB \to Rn \to T$  | 1                        | LSB | Yes  | Yes     |                        |
| ROTCL       | Rn     | 0100nnnn00100100 | $T \leftarrow Rn \leftarrow T$  | 1                        | MSB | Yes  | Yes     |                        |
| ROTCR       | Rn     | 0100nnnn00100101 | $T \to Rn \to T$  | 1                        | LSB | Yes  | Yes     |                        |
| SHAD        | Rm, Rn | 0100nnnnmmm1100  | When $Rm \ge 0$ , $Rn <<$<br>$Rm \to Rn$<br>When $Rm < 0$ , $Rn >>$<br>$IRmI \to$<br>$[MSB \to Rn]$ | 1                        | _   |      | Yes     |                        |

|             |        |                  |   |                          |     | C    | ompatib | ility                  |
|-------------|--------|------------------|---|--------------------------|-----|------|---------|------------------------|
| Instruction | on     | Instruction Code | Operation   | Execu-<br>tion<br>Cycles |     | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| SHAL        | Rn     | 0100nnnn00100000 | $T \leftarrow Rn \leftarrow 0$  | 1                        | MSB | Yes  | Yes     |                        |
| SHAR        | Rn     | 0100nnnn00100001 | $MSB \to Rn \to T$  | 1                        | LSB | Yes  | Yes     |                        |
| SHLD        | Rm, Rn | 0100nnnnmmm1101  | When $Rm \ge 0$ , $Rn <<$<br>$Rm \rightarrow Rn$<br>When $Rm < 0$ , $Rn >>$<br>$IRmI \rightarrow$<br>$[0 \rightarrow Rn]$ | 1                        | _   |      | Yes     |                        |
| SHLL        | Rn     | 0100nnnn00000000 | $T \leftarrow Rn \leftarrow 0$  | 1                        | MSB | Yes  | Yes     |                        |
| SHLR        | Rn     | 0100nnnn00000001 | $0 \to Rn \to T$  | 1                        | LSB | Yes  | Yes     |                        |
| SHLL2       | Rn     | 0100nnnn00001000 | $Rn \ll 2 \rightarrow Rn$   | 1                        | _   | Yes  | Yes     |                        |
| SHLR2       | Rn     | 0100nnnn00001001 | $Rn >> 2 \rightarrow Rn$  | 1                        | _   | Yes  | Yes     |                        |
| SHLL8       | Rn     | 0100nnnn00011000 | Rn << 8 → Rn  | 1                        | _   | Yes  | Yes     |                        |
| SHLR8       | Rn     | 0100nnnn00011001 | Rn >> 8 → Rn  | 1                        | _   | Yes  | Yes     |                        |
| SHLL16      | Rn     | 0100nnnn00101000 | $Rn \ll 16 \rightarrow Rn$  | 1                        | _   | Yes  | Yes     |                        |
| SHLR16      | Rn     | 0100nnnn00101001 | Rn >> 16 → Rn   | 1                        | _   | Yes  | Yes     |                        |

## 2.5.6 Branch Instructions

**Table 2.16 Branch Instructions** 

|          |       |                  |   |                                | C    | ompatik | oility                 |
|----------|-------|------------------|---|--------------------------------|------|---------|------------------------|
| Instruct | tion  | Instruction Code | Operation   | Execu-<br>tion<br>Cycles T Bit | SH2E | SH4     | SH-2A/<br>SH2A-<br>FPU |
| BF       | label | 10001011dddddddd | When T = 0, disp $\times$ 2 + PC $\rightarrow$ PC, When T = 1, nop                | 3/1* —                         | Yes  | Yes     |                        |
| BF/S     | label | 10001111dddddddd | Delayed branch When T = 0, disp $\times$ 2 + PC $\rightarrow$ PC, When T = 1, nop | 2/1* —                         | Yes  | Yes     |                        |
| ВТ       | label | 10001001dddddddd | When T = 1, disp $\times$ 2 + PC $\rightarrow$ PC, When T = 0, nop                | 3/1* —                         | Yes  | Yes     |                        |

|             |                  |                  |   |                          |   | Compatibility |     |                        |  |
|-------------|------------------|------------------|---|--------------------------|---|---------------|-----|------------------------|--|
| Instruction | on               | Instruction Code | Operation   | Execu-<br>tion<br>Cycles |   | SH2E          | SH4 | SH-2A/<br>SH2A-<br>FPU |  |
| BT/S        | label            | 10001101dddddddd | Delayed branch When T = 1, disp $\times$ 2 + PC $\rightarrow$ PC, When T = 0, nop | 2/1*                     | _ | Yes           | Yes |                        |  |
| BRA         | label            | 1010dddddddddddd | Delayed branch,<br>$disp \times 2 + PC \rightarrow PC$                            | 2                        | = | Yes           | Yes |                        |  |
| BRAF        | Rm               | 0000mmmm00100011 | Delayed branch,<br>Rm + PC → PC   | 2                        | _ | Yes           | Yes |                        |  |
| BSR         | label            | 1011dddddddddddd | Delayed branch, PC $\rightarrow$ PR, disp $\times$ 2 + PC $\rightarrow$ PC        | 2                        | _ | Yes           | Yes |                        |  |
| BSRF        | Rm               | 0000mmmm00000011 | Delayed branch, PC $\rightarrow$ PR, Rm + PC $\rightarrow$ PC                     | 2                        | _ | Yes           | Yes |                        |  |
| JMP         | @Rm              | 0100mmmm00101011 | Delayed branch, $Rm \rightarrow PC$   | 2                        | _ | Yes           | Yes |                        |  |
| JSR         | @Rm              | 0100mmmm00001011 | Delayed branch, PC $\rightarrow$ PR, Rm $\rightarrow$ PC                          | 2                        | _ | Yes           | Yes |                        |  |
| JSR/N       | @Rm              | 0100mmmm01001011 | $PC\text{-}2\toPR,Rm\toPC$  | 3                        | _ |               |     | Yes                    |  |
| JSR/N       | @ @ (disp8, TBR) | 10000011dddddddd | $PC-2 \rightarrow PR$ ,<br>$(disp \times 4 + TBR) \rightarrow PC$                 | 5                        | _ |               |     | Yes                    |  |
| RTS         |                  | 0000000000001011 | Delayed branch, $PR \rightarrow PC$   | 2                        | _ | Yes           | Yes |                        |  |
| RTS/N       |                  | 000000001101011  | $PR \rightarrow PC$   | 3                        | _ |               |     | Yes                    |  |
| RTV/N       | Rm               | 0000mmmm01111011 | $Rm \rightarrow R0, PR \rightarrow PC$  | 3                        | _ |               |     | Yes                    |  |
|             |                  |                  |   |                          |   |               |     |                        |  |

Note: \* One cycle when the program does not branch.

# 2.5.7 System Control Instructions

**Table 2.17 System Control Instructions** 

|            |           |                  |   |                          |       | Compatibility |     |                        |  |
|------------|-----------|------------------|---|--------------------------|-------|---------------|-----|------------------------|--|
| Instructio | n         | Instruction Code | Operation   | Execu-<br>tion<br>Cycles | T Bit | SH2E          | SH4 | SH-2A/<br>SH2A-<br>FPU |  |
| CLRT       |           | 0000000000001000 | 0 → T   | 1                        | 0     | Yes           | Yes |                        |  |
| CLRMAC     |           | 000000000101000  | 0 → MACH,MACL                                       | 1                        |       | Yes           | Yes |                        |  |
| LDBANK     | @Rm,R0    | 0100mmmm11100101 | (Specified register bank entry) → R0                | 6                        | _     |               |     | Yes                    |  |
| LDC        | Rm,SR     | 0100mmmm00001110 | $Rm \rightarrow SR$                                 | 3                        | LSB   | Yes           | Yes |                        |  |
| LDC        | Rm,TBR    | 0100mmmm01001010 | $Rm \rightarrow TBR$                                | 1                        | _     |               |     | Yes                    |  |
| LDC        | Rm,GBR    | 0100mmmm00011110 | $Rm \to GBR$  | 1                        | _     | Yes           | Yes |                        |  |
| LDC        | Rm,VBR    | 0100mmmm00101110 | $Rm \to VBR$  | 1                        | _     | Yes           | Yes |                        |  |
| LDC.L      | @Rm+,SR   | 0100mmmm00000111 | $(Rm) \rightarrow SR, Rm + 4 \rightarrow$ $Rm$      | 5                        | LSB   | Yes           | Yes |                        |  |
| LDC.L      | @Rm+,GBR  | 0100mmmm00010111 | $(Rm) \rightarrow GBR, Rm + 4 \rightarrow Rm$       | 1                        | _     | Yes           | Yes |                        |  |
| LDC.L      | @Rm+,VBR  | 0100mmmm00100111 | $(Rm) \rightarrow VBR, Rm + 4 \rightarrow Rm$       | 1                        | _     | Yes           | Yes |                        |  |
| LDS        | Rm,MACH   | 0100mmmm00001010 | $Rm \rightarrow MACH$                               | 1                        | _     | Yes           | Yes |                        |  |
| LDS        | Rm,MACL   | 0100mmmm00011010 | Rm 	o MACL  | 1                        | _     | Yes           | Yes |                        |  |
| LDS        | Rm,PR     | 0100mmmm00101010 | $Rm \rightarrow PR$                                 | 1                        | _     | Yes           | Yes |                        |  |
| LDS.L      | @Rm+,MACH | 0100mmmm00000110 | $(Rm) \rightarrow MACH, Rm + 4$<br>$\rightarrow Rm$ | 1                        | _     | Yes           | Yes |                        |  |
| LDS.L      | @Rm+,MACL | 0100mmmm00010110 | $(Rm) \rightarrow MACL, Rm + 4$<br>$\rightarrow Rm$ | 1                        | _     | Yes           | Yes |                        |  |
| LDS.L      | @Rm+,PR   | 0100mmmm00100110 | $(Rm) \rightarrow PR, Rm + 4 \rightarrow Rm$        | 1                        | _     | Yes           | Yes |                        |  |
| NOP        |           | 000000000001001  | No operation  | 1                        | _     | Yes           | Yes |                        |  |
| RESBANK    | (         | 000000001011011  | Bank → R0 to R14,<br>GBR,<br>MACH, MACL, PR         | 9*                       | _     |               |     | Yes                    |  |
| RTE        |           | 000000000101011  | Delayed branch,<br>stack area → PC/SR               | 6                        | _     | Yes           | Yes |                        |  |

|             |           |                  |  |                          |       | Compatibility |     |                        |
|-------------|-----------|------------------|--|--------------------------|-------|---------------|-----|------------------------|
| Instruction |           | Instruction Code | Operation  | Execu-<br>tion<br>Cycles | T Bit | SH2E          | SH4 | SH-2A/<br>SH2A-<br>FPU |
| SETT        |           | 000000000011000  | $1 \rightarrow T$  | 1                        | 1     | Yes           | Yes |                        |
| SLEEP       |           | 000000000011011  | Sleep  | 5                        | _     | Yes           | Yes |                        |
| STBANK      | R0,@Rn    | 0100nnnn11100001 | R0 → (specified register bank entry)                                     | 7                        | _     |               |     | Yes                    |
| STC         | SR,Rn     | 0000nnnn00000010 | $SR \rightarrow Rn$  | 2                        | _     | Yes           | Yes |                        |
| STC         | TBR,Rn    | 0000nnnn01001010 | $TBR \to Rn$   | 1                        | _     |               |     | Yes                    |
| STC         | GBR,Rn    | 0000nnnn00010010 | $GBR \rightarrow Rn$   | 1                        | _     | Yes           | Yes |                        |
| STC         | VBR,Rn    | 0000nnnn00100010 | $VBR \rightarrow Rn$   | 1                        |       | Yes           | Yes |                        |
| STC.L       | SR,@-Rn   | 0100nnnn00000011 | $Rn-4 \rightarrow Rn, SR \rightarrow (Rn)$                               | 2                        | _     | Yes           | Yes |                        |
| STC.L       | GBR,@-Rn  | 0100nnnn00010011 | $Rn-4 \rightarrow Rn, GBR \rightarrow$ (Rn)                              | 1                        | _     | Yes           | Yes |                        |
| STC.L       | VBR,@-Rn  | 0100nnnn00100011 | Rn-4 $\rightarrow$ Rn, VBR $\rightarrow$ (Rn)                            | 1                        | _     | Yes           | Yes |                        |
| STS         | MACH,Rn   | 0000nnnn00001010 | $MACH \rightarrow Rn$  | 1                        | _     | Yes           | Yes |                        |
| STS         | MACL,Rn   | 0000nnnn00011010 | $MACL \rightarrow Rn$  | 1                        |       | Yes           | Yes |                        |
| STS         | PR,Rn     | 0000nnnn00101010 | $PR \rightarrow Rn$  | 1                        | _     | Yes           | Yes |                        |
| STS.L       | MACH,@-Rn | 0100nnnn00000010 | $Rn-4 \rightarrow Rn, MACH \rightarrow$ (Rn)                             | 1                        | _     | Yes           | Yes |                        |
| STS.L       | MACL,@-Rn | 0100nnnn00010010 | $Rn-4 \rightarrow Rn, MACL \rightarrow$ (Rn)                             | 1                        | _     | Yes           | Yes |                        |
| STS.L       | PR,@-Rn   | 0100nnnn00100010 | $Rn-4 \rightarrow Rn, PR \rightarrow (Rn)$                               | 1                        | _     | Yes           | Yes |                        |
| TRAPA       | #imm      | 11000011iiiiiiii | $PC/SR \rightarrow stack area,$<br>$(imm \times 4 + VBR) \rightarrow PC$ | 5                        | _     | Yes           | Yes |                        |

Notes: Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states in cases such as the following:

- a. When there is a conflict between an instruction fetch and a data access
- b. When the destination register of a load instruction (memory  $\rightarrow$  register) is the same as the register used by the next instruction.
- \* In the event of bank overflow, the number of cycles is 19.

SH72546R Section 2 CPU

## 2.5.8 Floating-Point Operation Instructions

**Table 2.18 Floating-Point Operation Instructions** 

|             |             |                  |                                      |                          |                           | Compatibility |     | ility                  |
|-------------|-------------|------------------|--------------------------------------|--------------------------|---------------------------|---------------|-----|------------------------|
| Instruction | n           | Instruction Code | Operation                            | Execu-<br>tion<br>Cycles | T Bit                     | SH2E          | SH4 | SH-2A/<br>SH2A-<br>FPU |
| FABS        | FRn         | 1111nnnn01011101 | $ FRn  \rightarrow FRn$              | 1                        | _                         | Yes           | Yes |                        |
| FABS        | DRn         | 1111nnn001011101 | $ DRn  \to DRn$                      | 1                        | _                         |               | Yes |                        |
| FADD        | FRm, FRn    | 1111nnnnmmmm0000 | $FRn + FRm \rightarrow FRn$          | 1                        | _                         | Yes           | Yes |                        |
| FADD        | DRm, DRn    | 1111nnn0mmm00000 | $DRn + DRm \rightarrow DRn$          | 6                        | -                         |               | Yes |                        |
| FCMP/EQ     | FRm, FRn    | 1111nnnnmmmm0100 | (FRn = FRm)? 1:0 $\rightarrow$ T     | 1                        | Compa-<br>rison<br>result | Yes           | Yes |                        |
| FCMP/EQ     | DRm, DRn    | 1111nnn0mmm00100 | (DRn = DRm)? 1:0 → T                 | 2                        | Compa-<br>rison<br>result |               | Yes |                        |
| FCMP/GT     | FRm, FRn    | 1111nnnnmmmm0101 | (FRn > FRm)? 1:0 $\rightarrow$ T     | 1                        | Compa<br>-rison<br>result | Yes           | Yes |                        |
| FCMP/GT     | DRm, DRn    | 1111nnn0mmm00101 | (DRn > DRm)? $1:0 \rightarrow T$     | 2                        | Compa-<br>rison<br>result |               | Yes |                        |
| FCNVDS      | DRm, FPUL   | 1111mmm010111101 | (float) DRm $\rightarrow$ FPUL       | 2                        | _                         |               | Yes |                        |
| FCNVSD      | FPUL, DRn   | 1111nnn010101101 | (double) FPUL $\rightarrow$ DRn      | 2                        | _                         |               | Yes |                        |
| FDIV        | FRm, FRn    | 1111nnnnmmmm0011 | $FRn/FRm \to FRn$                    | 10                       | _                         | Yes           | Yes |                        |
| FDIV        | DRm, DRn    | 1111nnn0mmm00011 | $DRn/DRm \to DRn$                    | 23                       | _                         |               | Yes |                        |
| FLDI0       | FRn         | 1111nnnn10001101 | $0 \times 00000000 \rightarrow FRn$  | 1                        | _                         | Yes           | Yes |                        |
| FLDI1       | FRn         | 1111nnnn10011101 | $0 \times 3F800000 \rightarrow FRn$  | 1                        | _                         | Yes           | Yes |                        |
| FLDS        | FRm, FPUL   | 1111mmmm00011101 | $FRm \to FPUL$                       | 1                        | _                         | Yes           | Yes |                        |
| FLOAT       | FPUL,FRn    | 1111nnnn00101101 | $(float)FPUL \to FRn$                | 1                        | _                         | Yes           | Yes |                        |
| FLOAT       | FPUL,DRn    | 1111nnn000101101 | $(double)FPUL \to DRn$               | 2                        | _                         |               | Yes |                        |
| FMAC        | FR0,FRm,FRn | 1111nnnnmmmm1110 | $FR0 \times FRm+FRn \rightarrow FRn$ | 1                        | _                         | Yes           | Yes |                        |
| FMOV        | FRm, FRn    | 1111nnnnmmmm1100 | $FRm \rightarrow FRn$                | 1                        | -                         | Yes           | Yes |                        |
| FMOV        | DRm, DRn    | 1111nnn0mmm01100 | $DRm \to DRn$                        | 2                        | -                         |               | Yes |                        |

Section 2 CPU SH72546R

|                     |                  |                                      |  |                          |   | Compatibil |     | ility                  |
|---------------------|------------------|--------------------------------------|--|--------------------------|---|------------|-----|------------------------|
| Instructio          | n                | Instruction Code                     | Operation                              | Execu-<br>tion<br>Cycles |   | SH2E       | SH4 | SH-2A/<br>SH2A-<br>FPU |
| FMOV.S              | @(R0, Rm), FRn   | 1111nnnnmmmm0110                     | $(R0 + Rm) \rightarrow FRn$            | 1                        | _ | Yes        | Yes |                        |
| FMOV.D              | @(R0, Rm), DRn   | 1111nnn0mmmm0110                     | $(R0 + Rm) \rightarrow DRn$            | 2                        |   |            | Yes |                        |
| FMOV.S              | @Rm+, FRn        | 1111nnnnmmmm1001                     | $(Rm) \rightarrow FRn, Rm+=4$          | 1                        | _ | Yes        | Yes |                        |
| FMOV.D              | @Rm+, DRn        | 1111nnn0mmmm1001                     | (Rm) → DRn, Rm += 8                    | 2                        | _ |            | Yes |                        |
| FMOV.S              | @Rm, FRn         | 1111nnnnmmm1000                      | $(Rm) \rightarrow FRn$                 | 1                        | _ | Yes        | Yes |                        |
| FMOV.D              | @Rm, DRn         | 1111nnn0mmmm1000                     | $(Rm) \rightarrow DRn$                 | 2                        | _ |            | Yes |                        |
| FMOV.S              | @(disp12,Rm),FRn | 0011nnnnmmmm0001                     | $(disp \times 4 + Rm) \to FRn$         | 1                        | _ |            |     | Yes                    |
|                     |                  | 0111dddddddddddd                     |  |                          |   |            |     |                        |
| FMOV.D              | @(disp12,Rm),DRn | 0011nnn0mmmm0001                     | $(disp \times 8 + Rm) \to DRn$         | 2                        | _ |            |     | Yes                    |
|                     |                  | 0111dddddddddddd                     |  |                          |   |            |     |                        |
| FMOV.S              | FRm, @(R0,Rn)    | 1111nnnnmmmm0111                     | $FRm \rightarrow (R0 + Rn)$            | 1                        | _ | Yes        | Yes |                        |
| FMOV.D              | DRm, @(R0,Rn)    | 1111nnnnmmm00111                     | $DRm \to (R0 + Rn)$                    | 2                        | _ |            | Yes |                        |
| FMOV.S              | FRm, @-Rn        | 1111nnnnmmmm1011                     | Rn-=4, FRm $\rightarrow$ (Rn)          | 1                        | _ | Yes        | Yes |                        |
| FMOV.D              | DRm, @-Rn        | 1111nnnnmmm01011                     | Rn-=8, DRm $\rightarrow$ (Rn)          | 2                        | _ |            | Yes |                        |
| FMOV.S              | FRm, @Rn         | 1111nnnnmmmm1010                     | $FRm \rightarrow (Rn)$                 | 1                        | _ | Yes        | Yes |                        |
| FMOV.D              | DRm, @Rn         | 1111nnnnmmm01010                     | $DRm \to (Rn)$                         | 2                        | _ |            | Yes |                        |
| FMOV.S<br>@(disp12, | FRm,<br>Rn)      | 0011nnnnmmmm0001<br>0011dddddddddddd | $FRm \rightarrow (disp \times 4 + Rn)$ | 1                        | _ |            |     | Yes                    |
| FMOV.D<br>@(disp12, | DRm,<br>Rn)      | 0011nnnnmmm00001<br>0011dddddddddddd | $DRm \to (disp \times 8 + Rn)$         | 2                        | _ |            |     | Yes                    |
| FMUL                | FRm, FRn         | 1111nnnnmmmm0010                     | $FRn \times FRm \rightarrow FRn$       | 1                        |   | Yes        | Yes |                        |
| FMUL                | DRm, DRn         | 1111nnn0mmm00010                     | $DRn \times DRm \to DRn$               | 6                        | _ |            | Yes |                        |
| FNEG                | FRn              | 1111nnnn01001101                     | -FRn → FRn                             | 1                        | _ | Yes        | Yes |                        |
| FNEG                | DRn              | 1111nnn001001101                     | -DRn → DRn                             | 1                        | _ |            | Yes |                        |
| FSCHG               |                  | 11110011111111101                    | FPSCR.SZ=~FPSCR.S                      | 1                        |   |            | Yes |                        |
| FSQRT               | FRn              | 1111nnnn01101101                     | $\sqrt{FRn} \rightarrow FRn$           | 9                        | _ |            | Yes |                        |
| FSQRT               | DRn              | 1111nnn001101101                     | $\sqrt{DRn} \to DRn$                   | 22                       | _ |            | Yes |                        |
| FSTS                | FPUL,FRn         | 1111nnnn00001101                     | $FPUL \to FRn$                         | 1                        | _ | Yes        | Yes |                        |
| FSUB                | FRm, FRn         | 1111nnnnmmmm0001                     | $FRn-FRm \rightarrow FRn$              | 1                        | _ | Yes        | Yes |                        |
|                     |                  |                                      |  |                          |   |            |     |                        |

SH72546R Section 2 CPU

|             |           |                  |                              |                          |   | Co   | ility |                        |
|-------------|-----------|------------------|------------------------------|--------------------------|---|------|-------|------------------------|
| Instruction | on        | Instruction Code | Operation                    | Execu-<br>tion<br>Cycles |   | SH2E | SH4   | SH-2A/<br>SH2A-<br>FPU |
| FSUB        | DRm, DRn  | 1111nnn0mmm00001 | $DRn\text{-}DRm\toDRn$       | 6                        | _ |      | Yes   |                        |
| FTRC        | FRm, FPUL | 1111mmmm00111101 | $(long)FRm \rightarrow FPUL$ | 1                        | _ | Yes  | Yes   |                        |
| FTRC        | DRm, FPUL | 1111mmm000111101 | $(long)DRm\toFPUL$           | 2                        | _ |      | Yes   |                        |

### 2.5.9 FPU-Related CPU Instructions

**Table 2.19 FPU-Related CPU Instructions** 

|            |             |                  |                                   |                          |   | Co   | Compatibility |                        |
|------------|-------------|------------------|-----------------------------------|--------------------------|---|------|---------------|------------------------|
| Instructio | n           | Instruction Code | Operation                         | Execu-<br>tion<br>Cycles |   | SH2E | SH4           | SH-2A/<br>SH2A-<br>FPU |
| LDS        | Rm,FPSCR    | 0100mmmm01101010 | $Rm \to FPSCR$                    | 1                        | _ | Yes  | Yes           |                        |
| LDS        | Rm,FPUL     | 0100mmmm01011010 | Rm 	o FPUL                        | 1                        | _ | Yes  | Yes           |                        |
| LDS.L      | @Rm+, FPSCR | 0100mmmm01100110 | $(Rm) \rightarrow FPSCR, Rm+=4$   | 1                        | _ | Yes  | Yes           |                        |
| LDS.L      | @Rm+, FPUL  | 0100mmmm01010110 | $(Rm) \rightarrow FPUL,  Rm += 4$ | 1                        | _ | Yes  | Yes           |                        |
| STS        | FPSCR, Rn   | 0000nnnn01101010 | $FPSCR \to Rn$                    | 1                        | _ | Yes  | Yes           |                        |
| STS        | FPUL,Rn     | 0000nnnn01011010 | $FPUL \to Rn$                     | 1                        | _ | Yes  | Yes           |                        |
| STS.L      | FPSCR,@-Rn  | 0100nnnn01100010 | Rn-=4, FPCSR $\rightarrow$ (Rn)   | 1                        | _ | Yes  | Yes           |                        |
| STS.L      | FPUL,@-Rn   | 0100nnnn01010010 | Rn-=4, FPUL $\rightarrow$ (Rn)    | 1                        | = | Yes  | Yes           |                        |

## 2.5.10 Bit Manipulation Instructions

**Table 2.20 Bit Manipulation Instructions** 

|                          |                                      |  |                |                          | Compatibility |     | ility           |
|--------------------------|--------------------------------------|--|----------------|--------------------------|---------------|-----|-----------------|
|                          |                                      |  | Execu-<br>tion |                          |               |     | SH-2A/<br>SH2A- |
| Instruction              | Instruction Code                     | Operation                                | Cycles         | T Bit                    | SH2E          | SH4 | FPU             |
| BAND.B#imm3,@(disp12,Rn) | 0011nnnn0iii1001<br>0100dddddddddddd | (imm of (disp + Rn)) & T $\rightarrow$ T | 3              | Ope-<br>ration<br>result |               |     | Yes             |

Section 2 CPU SH72546R

|         |                            |                                       |   |                          |                          | Co   | Compatibility |                        |  |
|---------|----------------------------|---------------------------------------|---|--------------------------|--------------------------|------|---------------|------------------------|--|
| Instruc | tion                       | Instruction Code                      | Operation                                   | Execu-<br>tion<br>Cycles | T Bit                    | SH2E | SH4           | SH-2A/<br>SH2A-<br>FPU |  |
| BANDN   | OT.B<br>#imm3,@(disp12,Rn) | 0011nnnn0iii1001<br>1100dddddddddddd  | ~(imm of (disp + Rn)) & $T \rightarrow T$   | 3                        | Ope-<br>ration<br>result |      |               | Yes                    |  |
| BCLR.E  | 3 #imm3,@(disp12,Rn)       | 0011nnnn0iii1001<br>0000dddddddddddd  | $0 \rightarrow \text{(imm of (disp + Rn))}$ | 3                        | _                        |      |               | Yes                    |  |
| BCLR    | #imm3,Rn                   | 10000110nnnn0iii                      | 0 → imm of Rn                               | 1                        | _                        |      |               | Yes                    |  |
| BLD.B   | #imm3,@(disp12,Rn)         | 0011nnnn0iii1001<br>0011ddddddddddddd | $(imm\;of\;(disp+Rn))\to$                   | 3                        | Ope-<br>ration<br>result |      |               | Yes                    |  |
| BLD     | #imm3,Rn                   | 10000111nnnn1iii                      | imm of Rn $\rightarrow$ T                   | 1                        | Ope-<br>ration<br>result |      |               | Yes                    |  |
| BLDNO   | T.B<br>#imm3,@(disp12,Rn)  | 0011nnnn0iii1001<br>1011ddddddddddddd | ~(imm of (disp + Rn))<br>→ T                | 3                        | Ope-<br>ration<br>result |      |               | Yes                    |  |
| BOR.B   | #imm3,@(disp12,Rn)         | 0011nnnn0iii1001<br>0101ddddddddddddd | ( imm of (disp + Rn))   T $\rightarrow$ T   | 3                        | Ope-<br>ration<br>result |      |               | Yes                    |  |
| BORNO   | DT.B<br>#imm3,@(disp12,Rn) | 0011nnnn0iii1001<br>1101ddddddddddddd | ~( imm of (disp + Rn))   $T \rightarrow T$  | 3                        | Ope-<br>ration<br>result |      |               | Yes                    |  |
| BSET.E  | 3<br>#imm3,@(disp12,Rn)    | 0011nnnn0iii1001<br>0001dddddddddddd  | $1 \rightarrow \text{(imm of (disp + Rn))}$ | 3                        | _                        |      |               | Yes                    |  |
| BSET    | #imm3,Rn                   | 10000110nnnn1iii                      | $1 \rightarrow \text{imm of Rn}$            | 1                        | _                        |      |               | Yes                    |  |
| BST.B   | #imm3,@(disp12,Rn)         | 0011nnnn0iii1001<br>0010dddddddddddd  | $T \rightarrow \text{(imm of (disp + Rn))}$ | 3                        | _                        |      |               | Yes                    |  |
| BST     | #imm3,Rn                   | 10000111nnnn0iii                      | $T \rightarrow imm \ of \ Rn$               | 1                        | _                        |      |               | Yes                    |  |
| BXOR.   | 3<br>#imm3,@(disp12,Rn)    | 0011nnnn0iii1001<br>0110dddddddddddd  | (imm of (disp + Rn)) $^T$ $\rightarrow$ T   | 3                        | Ope-<br>ration<br>result |      |               | Yes                    |  |

SH72546R Section 2 CPU

## 2.6 Processing States

The CPU has four processing states: reset, exception handling, program execution, and power-down. Figure 2.8 shows the transitions between the states.

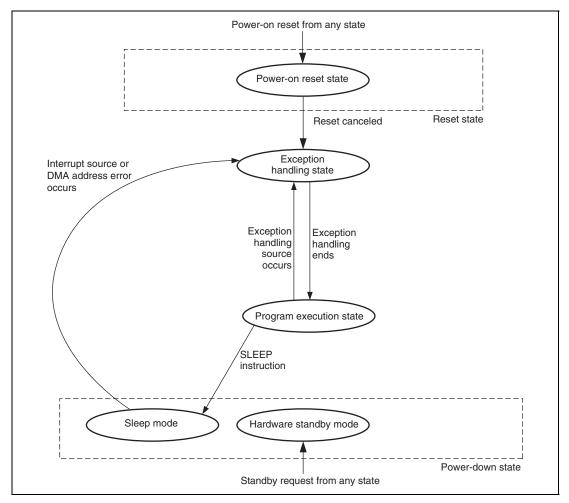


Figure 2.8 Transitions between Processing States

### (1) Reset State

In this state, the CPU is reset by a power-on reset.

Section 2 CPU SH72546R

#### (2) Exception Handling State

The exception handling state is a transient state that occurs when exception handling sources such as resets or interrupts alter the CPU's processing state flow.

For a reset, the initial values of the program counter (PC) (execution start address) and stack pointer (SP) are fetched from the exception handling vector table and stored; the CPU then branches to the execution start address and execution of the program begins.

For an interrupt, the stack pointer (SP) is accessed and the program counter (PC) and status register (SR) are saved to the stack area. The exception service routine start address is fetched from the exception handling vector table; the CPU then branches to that address and the program starts executing, thereby entering the program execution state.

#### (3) Program Execution State

In the program execution state, the CPU sequentially executes the program.

#### (4) Power-Down State

In the power-down state, the CPU stops operating to conserve power. Sleep mode is entered by executing a SLEEP instruction. If hardware standby input is received, the CPU enters the hardware standby mode.

## Section 3 Operating Modes

This LSI has a JTAG interface, so the TRST signal must be held low for a specified time when power is supplied. This is the case whether or not the JTAG interface is used, even in normal operating mode.

## 3.1 Types of Operating Modes and Selection

This LSI has six types of operating modes. The operating mode is determined by the setting of pins MD4 to MD0 and FWE. The setting should not be changed during LSI operation. However, the shift of mode from the MCU extended mode with the on-chip ROM enabled or the MCU single-chip mode to the user program mode is supported even when the FWE pin is operating. These pins should be set as shown in table 3.2.

The voltage of the PVcc1 power should range within the values shown in table 3.1.

**Table 3.1 Selection of Operating Modes** 

| Mode No. | Mode Name            | On-Chip<br>ROM | External<br>Bus | Bus Width<br>(Area 0) | Write-<br>Access to<br>ROM | PVcc1<br>Voltage |
|----------|----------------------|----------------|-----------------|-----------------------|----------------------------|------------------|
| Mode 0   | MCU extended mode    | Disabled       | Enabled         | 8 bits                | _                          | 3.3 V ±0.3 V     |
| Mode 1   | _                    | Disabled       | Enabled         | 16 bits               | _                          | 3.3 V ±0.3 V     |
| Mode 2   | _                    | Enabled        | Enabled         | Set by<br>CS0BCR*     | Disabled                   | 3.3 V ±0.3 V     |
| Mode 3   | MCU single-chip mode | Enabled        | Disabled        | _                     | Disabled                   | 5.0 V ±0.5 V     |
| Mode 4   | Boot mode            | Enabled        | Enabled         | Set by<br>CS0BCR*     | Enabled                    | 3.3 V ±0.3 V     |
| Mode 5   | _                    | Enabled        | Disabled        | _                     | Enabled                    | 5.0 V ±0.5 V     |
| Mode 6   | User program mode    | Enabled        | Enabled         | Set by<br>CS0BCR*     | Enabled                    | 3.3 V ±0.3 V     |
| Mode 7   | _                    | Enabled        | Disabled        | _                     | Enabled                    | 5.0 V ±0.5 V     |
| Mode 8   | User boot mode       | Enabled        | Enabled         | Set by<br>CS0BCR*     | Enabled                    | 3.3 V ±0.3 V     |
| Mode 9   | <del>-</del>         | Enabled        | Disabled        | _                     | Enabled                    | 5.0 V ±0.5 V     |

Note: \* The CSn space bus control register (CS0BCR) is a register of the bus state controller (BSC). The bus width can be selected from 8 bits or 16 bits.

**Table 3.2** Operating Mode Pin Settings

|          |                      |              |     | riii Sett | iliy |     |   |
|----------|----------------------|--------------|-----|-----------|------|-----|---|
| Mode No. | Mode Name            | MD4/MD3*1    | MD2 | MD1       | MD0  | FWE |   |
| Mode 0   | MCU extended mode    | 0            | 1   | 1         | 1    | 0*2 | _ |
| Mode 1   | <del>-</del>         |              | 1   | 1         | 1    | 1*2 | _ |
| Mode 2   | _                    |              | 0   | 0         | 1    | 0   | _ |
| Mode 3   | MCU single-chip mode | <del>_</del> | 0   | 0         | 0    | 0   | _ |
| Mode 4   | Boot mode            |              | 0   | 1         | 1    | 1   | _ |
| Mode 5   | _                    |              | 0   | 1         | 0    | 1   | _ |
| Mode 6   | User program mode    |              | 0   | 0         | 1    | 1   | _ |
| Mode 7   | _                    |              | 0   | 0         | 0    | 1   | _ |
| Mode 8   | User boot mode       | _            | 1   | 0         | 1    | 1   | _ |
| Mode 9   | _                    |              | 1   | 0         | 0    | 1   | _ |

Pin Setting

Notes: 1. Pins MD4 and MD3 should always be driven to a level of 0.

2. The FWE pin functions as a mode setting pin that is used to select the external bus width in on-chip ROM disabled mode.

There are two modes as the MCU operating modes: the MCU extended mode and single-chip mode.

There are three modes as the flash memory programming modes: on-board programming modes (boot mode, user boot mode, and user program mode).

Independent from the above operating modes, this LSI features an ASE mode, which allows debugging operations through the external connection of an emulator. Setting the ASEMD pin to a level of 1 makes the LSI the ASE mode. If, for example, the MCU single-chip mode is selected through the settings of pins MD4 to MD0 and FWE, this LSI operates in MCU single chip mode of ASE mode.

If the LSI is set in ASE mode without connecting an emulator, the correct operation cannot be guaranteed. If this LSI is not connected to an emulator, the ASEMD pin must be set to a level of 0 so that this LSI operates in normal operating mode.

Unless otherwise noted, descriptions given in this manual assume that this LSI operates in normal operating mode.

## Section 4 Clock Pulse Generator (CPG)

#### 4.1 Overview

The clock pulse generator (CPG) supplies clock pulses to both the inside of this LSI and external devices. The CPG consists of an oscillation circuit and a PLL multiplier. There are two methods of generating a clock with the CPG: by connecting a crystal resonator or by inputting an external clock.

The oscillation circuit oscillates at the same frequency as the input clock.

Two types of clock signals are internally supplied: the internal clock  $(\phi)$  and peripheral clock  $(P\phi)$ . The internal clock  $(\phi)$  signal is supplied to the modules such as the CPU, FPU, on-chip RAM, and ROM cache. The frequency of this clock is selected from four, six, eight, or ten times the frequency of the clock signal input on the EXTAL pin. The multiplication ratio can only be changed through the settings of pins MD\_CLK1 and MD\_CLK0 and cannot be changed during LSI operation.

The peripheral clock signal ( $P\phi$ ) is mainly supplied to the on-chip peripheral modules. The frequency of this clock is selected from one or two times the frequency of the input clock from the EXTAL pin. The multiplication ratio can only be changed by setting MD\_CLKP and cannot be changed during LSI operation. The CK pin outputs the peripheral clock signal ( $P\phi$ ).

The CPG is halted in hardware standby mode.

A block diagram of the CPG is shown in figure 4.1.

Apr 01, 2014

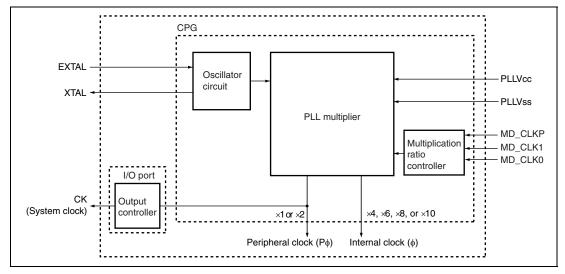


Figure 4.1 Block Diagram of CPG

#### 4.2 **Pin Configuration**

Table 4.1 shows the pin configuration of the CPG.

Table 4.1 **Pin Configuration** 

| Pin Name           | Symbol   | I/O    | Function  |
|--------------------|--|--------|---|
| External clock     | EXTAL  | Input  | Input pin for crystal resonator or external clock |
| Crystal            | XTAL   | Output | Input pin for crystal resonator                   |
| System clock       | CK   | Output | Output pin for system clock                       |
| Clock mode setting | g MD_CLKP Input Input pin for setting peripheral |        | Input pin for setting peripheral clock frequency  |
|                    | MD_CLK0,<br>MD_CLK1                              | Input  | Input pins for setting PLL multiplication ratio   |
| PLL power supply   | PLLV <sub>cc</sub>                               | Input  | Power supply pin for PLL multiplier               |
| PLL ground         | PLLV <sub>ss</sub>                               | Input  | Ground pin for PLL multiplier                     |

## 4.3 Frequency Ranges and Clock Selection

Two types of clock signals, internal clock ( $\phi$ ) and peripheral clock ( $P\phi$ ) signals, are internally supplied.

This clock is supplied to the modules such as the CPU, FPU, on-chip RAM, and ROM cache. The frequency of the internal clock ( $\phi$ ) signal is selected from four, six, eight, or ten times the frequency of the clock signal input on the EXTAL pin according to the settings of pins MD\_CLK1 and MD\_CLK0.

The peripheral clock signal  $(P\phi)$  is mainly supplied to the on-chip peripheral modules and is selected from one or two times the input frequency on the EXTAL pin according to the settings of the MD\_CLKP pin. The CK pin outputs the peripheral clock signal  $(P\phi)$ .

The input frequency and operating frequency ranges for each pin setting are shown in table 4.2.

The CK pin enables or disables the pin output through the setting of the CK control register (CKCR) of the I/O port. For details on CKCR, see section 23, I/O Ports.

**Table 4.2** Input Frequency and Operating Frequency

| Pin Setting |         |         | Input                    | PLL                      | Internal Clock           | •                                 |  |
|-------------|---------|---------|--------------------------|--------------------------|--------------------------|-----------------------------------|--|
| MD_CLKP     | MD_CLK1 | MD_CLK0 | Frequency<br>Range (MHz) | Multiplication<br>Factor | Frequency<br>Range (MHz) | Clock<br>Frequency<br>Range (MHz) |  |
| 0           | 0       | 0       | 16 to 20                 | × 4                      | 64 to 80                 | 16 to 20                          |  |
|             |         | 1       | -                        | × 6                      | 96 to 120                | -                                 |  |
|             | 1       | 0       | -                        | × 10                     | 160 to 200               | <del>.</del>                      |  |
|             |         | 1       | -                        | × 8                      | 128 to 160               | <del>.</del>                      |  |
| 1           | 0       | 0       | -                        | × 4                      | 64 to 80                 | 32 to 40                          |  |
|             |         | 1       | -                        | × 6                      | 96 to 120                | -                                 |  |
|             | 1       | 0       | -                        | × 10                     | 160 to 200               | -                                 |  |
|             | 1       |         | -                        | × 8                      | 128 to 160               | -                                 |  |

Note: The multiplication ratio and pin settings must not be changed during operation of this LSI.

#### 4.4 Clock Source

A crystal resonator or an external clock can be selected as the clock source.

## 4.4.1 Connecting Crystal Resonator

#### (1) Circuit Configuration

Figure 4.2 shows an example of connecting a crystal resonator. Use the damping resistor (Rd) shown in table 4.3. An AT-cut parallel-resonance type crystal resonator should be used. Load capacitors (CL1, CL2) must be connected as shown in figure 4.2.

The clock pulses generated by the crystal resonator and internal oscillation circuit are sent to the PLL multiplier, where the clock signals are multiplied to produce the selected frequency and supplied internally and externally.

The crystal resonator manufacturer should be consulted concerning the compatibility between the crystal resonator and this LSI.

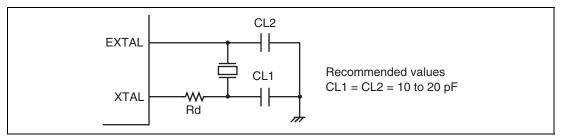


Figure 4.2 Connection Example of Crystal Resonator

**Table 4.3 Damping Resistor Values (Recommended Values)** 

|                | Frequency (MHz) |    |  |  |  |
|----------------|-----------------|----|--|--|--|
| Parameter      | 16              | 20 |  |  |  |
| $Rd\;(\Omega)$ | 0               | 0  |  |  |  |

### (2) Crystal Resonator

Figure 4.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator with the characteristics listed in table 4.4.

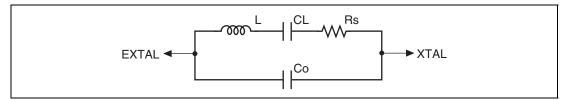


Figure 4.3 Crystal Resonator Equivalent Circuit

 Table 4.4
 Crystal Resonator Parameters (Recommended Values)

|           | Frequency (MHz) |    |   |  |  |
|-----------|-----------------|----|---|--|--|
| Parameter | 16              | 20 |   |  |  |
| Rs (Ω)    | 22              | 20 |   |  |  |
| Co (pF)   | 1               | 1  | _ |  |  |

### 4.4.2 External Clock Input

An example of external clock input connection is shown in figure 4.4.

Leave the XTAL pin open-circuit, but ensure that the parasitic capacitance on the XTAL pin is not greater than 1 pF.

Even when an external clock is input, secure the oscillation stabilization time when switching on or leaving the standby mode for PLL stabilization.

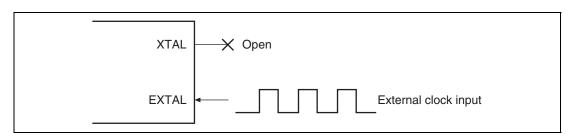


Figure 4.4 External Clock Input

## 4.5 Usage Notes

## 4.5.1 Note on Board Design

Place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

To prevent induction from interfering with correct oscillation, do not allow any signal lines to cross the XTAL or EXTAL lines (figure 4.5).

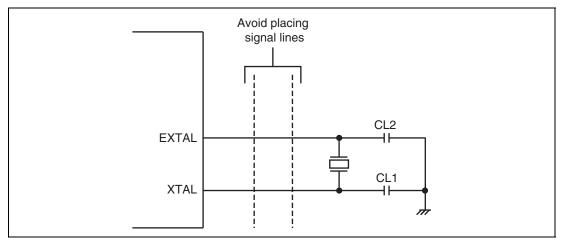


Figure 4.5 Note on Board Design

#### 4.5.2 Note on Connecting Power Supply for PLL Oscillator

Separate  $PLLV_{cc}$  and  $PLLV_{ss}$  from the other  $V_{cc}$  and  $V_{ss}$  lines at the board power supply source, and be sure to insert bypass capacitors  $C_{PB}$  and  $C_{B}$  close to the pins.

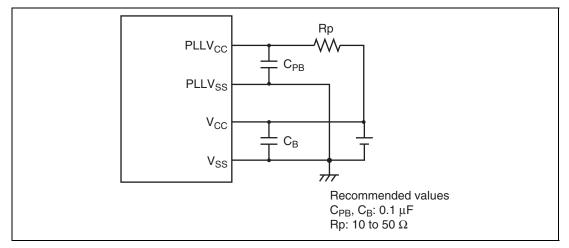


Figure 4.6 Note on Connecting Power Supply for PLL Oscillator

## Section 5 Address Space

Table 5.1 shows the address space of the SH72546RFCC in on-chip ROM enabled mode (exclusive of the single-chip mode); table 5.2 shows the address space of the SH72546RFCC in on-chip ROM disabled mode; and table 5.3 shows the address space of the SH72546RFCC in single-chip mode.

Access to the internal I/O register spaces can be made via the addresses shown in section 31, List of Registers. Access should not be made to any other addresses than those provided in the list and to reserved areas; otherwise, operation cannot be guaranteed.

Table 5.1 Address Space of the SH72546RFCC in On-Chip ROM Enabled Mode (Exclusive of Single-Chip Mode)

| Address  | Type of address space  | Size                       |
|--|--|----------------------------|
| H'0000 0000 to H'003B FFFF<br>(H'0000 0000 to H'0000 7FFF)* <sup>1</sup> | On-chip ROM (read from user MAT)<br>(on-chip ROM (read from user boot<br>MAT)) | 3.75 Mbytes<br>(32 Kbytes) |
| H'003C 0000 to H'0040 1FFF   | Reserved area  |                            |
| H'0040 2000 to H'0040 3FFF   | FCU firmware area  | 8 Kbytes                   |
| H'0040 4000 to H'01FF FFFF   | Reserved area  |                            |
| H'0200 0000 to H'023F FFFF   | External address space (CS0)   | 4 Mbytes                   |
| H'0240 0000 to H'03FF FFFF   | CS0 shadow space   | 28 Mbytes                  |
| H'0400 0000 to H'043F FFFF   | External address space (CS1)   | 4 Mbytes                   |
| H'0440 0000 to H'07FF FFFF   | CS1 shadow space   | 60 Mbytes                  |
| H'0800 0000 to H'083F FFFF   | External address space (CS2)   | 4 Mbytes                   |
| H'0840 0000 to H'0BFF FFFF   | CS2 shadow space   | 60 Mbytes                  |
| H'0C00 0000 to H'0C3F FFFF   | External address space (CS3)   | 4 Mbytes                   |
| H'0C40 0000 to H'0FFF FFFF   | CS3 shadow space   | 60 Mbytes                  |
| H'1000 0000 to H'800F FFFF   | Reserved area  |                            |
| H'8010 0000 to H'8011 FFFF   | EEPROM (read/write)*2  | 128 Kbytes                 |
| H'8012 0000 to H'807F FFFF   | Reserved area  |                            |
| H'8080 0000 to H'80BB FFFF<br>(H'8080 0000 to H'8080 7FFF)* <sup>1</sup> | On-chip ROM (write to user MAT) (on-chip ROM (write to user boot MAT))         | 3.75 Mbytes<br>(32 Kbytes) |
| H'80BC 0000 to H'80FF 7FFF   | Reserved area  |                            |
| H'80FF 8000 to H'80FF 9FFF   | FCU RAM area   | 8 Kbytes                   |
|  |  |                            |

| Address                    | Type of address space Size |                   |
|----------------------------|----------------------------|-------------------|
| H'80FF A000 to H'FFF7 FFFF | Reserved area              |                   |
| H'FFF8 0000 to H'FFFB FFFF | On-chip RAM                | 256 Kbytes        |
| H'FFFC 0000 to H'FFFF FFFF | Internal I/O register      | 256 Kbytes (Max.) |

Notes: 1. When the user boot MAT of the on-chip ROM is selected. See section 25, ROM.

2. The EEPROM area includes the lot trace information. See section 26, EEPROM.

Table 5.2 Address Space of the SH72546RFCC in On-Chip ROM Disabled Mode

| Address                    | Type of address space        | Size              |
|----------------------------|------------------------------|-------------------|
| H'0000 0000 to H'003F FFFF | External address space (CS0) | 4 Mbytes          |
| H'0040 0000 to H'03FF FFFF | CS0 shadow space             | 60 Mbytes         |
| H'0400 0000 to H'043F FFFF | External address space (CS1) | 4 Mbytes          |
| H'0440 0000 to H'07FF FFFF | CS1 shadow space             | 60 Mbytes         |
| H'0800 0000 to H'083F FFFF | External address space (CS2) | 4 Mbytes          |
| H'0840 0000 to H'0BFF FFFF | CS2 shadow space             | 60 Mbytes         |
| H'0C00 0000 to H'0C3F FFFF | External address space (CS3) | 4 Mbytes          |
| H'0C40 0000 to H'0FFF FFFF | CS3 shadow space             | 60 Mbytes         |
| H'1000 0000 to H'FFF7 FFFF | Reserved area                |                   |
| H'FFF8 0000 to H'FFFB FFFF | On-chip RAM                  | 256 Kbytes        |
| H'FFFC 0000 to H'FFFF FFFF | Internal I/O register        | 256 Kbytes (Max.) |

Table 5.3 Address Space of the SH72546RFCC in Single-Chip Mode

| Address  | Type of address space  | Size                       |
|--|--|----------------------------|
| H'0000 0000 to H'003B FFFF<br>(H'0000 0000 to H'0000 7FFF)* <sup>1</sup> | On-chip ROM (read from user MAT)<br>(on-chip ROM (read from user boot<br>MAT)) | 3.75 Mbytes<br>(32 Kbytes) |
| H'003C 0000 to H'0040 1FFF   | Reserved area  |                            |
| H'0040 2000 to H'0040 3FFF   | FCU firmware area  | 8 Kbytes                   |
| H'0040 4000 to H'800F FFFF   | Reserved area  |                            |
| H'8010 0000 to H'8011 FFFF   | EEPROM (read/write)*2  | 128 Kbytes                 |
| H'8012 0000 to H'807F FFFF   | Reserved area  | _                          |
| H'8080 0000 to H'80BB FFFF<br>(H'8080 0000 to H'8080 7FFF)* <sup>1</sup> | On-chip ROM (write to user MAT) (on-chip ROM (write to user boot MAT))         | 3.75 Mbytes<br>(32 Kbytes) |
| H'80BC 0000 to H'80FF 7FFF   | Reserved area  | _                          |
| H'80FF 8000 to H'80FF 9FFF   | FCU RAM area   | 8 Kbytes                   |
| H'80FF A000 to H'FFF7 FFFF   | Reserved area  | _                          |
| H'FFF8 0000 to H'FFFB FFFF   | On-chip RAM  | 256 Kbytes                 |
| H'FFFC 0000 to H'FFFF FFFF   | Internal I/O register  | 256 Kbytes (Max.)          |

Notes: 1. When the user boot MAT of the on-chip ROM is selected. See section 25, ROM.

2. The EEPROM area includes the lot trace information. See section 26, EEPROM.

SH72546R Section 6 Reset

## Section 6 Reset

When the  $\overline{RES}$  pin is driven low, the LSI enters the power-on reset state. If the  $\overline{RES}$  pin is driven high while the LSI is placed in power-on reset state, the power-on reset state is cancelled and the CPU starts power-on reset exception handling.

## 6.1 Reset Operation

When the  $\overline{RES}$  pin is driven low by the low level pulse longer than or equal to the noise cancellation width ( $t_{RESNCW}$ ), the reset request is accepted. When the reset request is accepted, all pin states are reset and each pin enters reset state. Each pin state during reset is summarized in appendix A, Pin States.

Internal circuits including the CPU are reset 3 to 4 P $\phi$  cycles after reset acceptance.

The  $\overline{RES}$  pin should be kept at the low level during at least  $t_{RESW}$  ( $t_{cyc}$ ). Later, if the  $\overline{RES}$  pin is driven high while it is kept low, the reset state of internal circuits is cancelled after 3 to 4 P $\phi$  cycles and the CPU starts power-on reset exception handling.

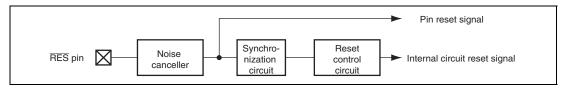


Figure 6.1 Reset Circuit

Section 6 Reset SH72546R

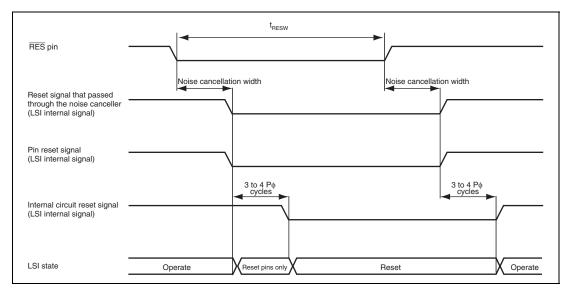


Figure 6.2 Reset Sequence

#### 6.1.1 Reset at Power-On

To reset at power-on, the  $\overline{\text{RES}}$  pin should be kept at the low level for the duration of the internal PLL oscillation settling time after the power supply voltage falls within the specified range and the  $\overline{\text{HSTBY}}$  pin is driven high. The internal PLL oscillation settling time is specified as  $t_{\text{osci}}$ . For details, see section 32, Electrical Characteristics.

## 6.1.2 Reset during Operation

To reset during operation, the RES pin should be kept at the low level for at least the reset pulse width. The reset pulse width is specified as  $t_{RESW}$ . For details, see section 32, Electrical Characteristics.

## 6.1.3 On-Chip RAM Data Retention during Reset

When the  $\overline{RES}$  pin is driven low causing a power-on reset while the internal bus master such as CPU accesses the RAM, the data at the accessed address may be lost. Data in addresses not accessed can be retained. To reliably retain the RAM data, the RAM should be disabled by the RAM enable register (RAMEN) before the  $\overline{RES}$  pin is driven low. For details on the RAM enable register, see section 28, RAM.

SH72546R Section 6 Reset

## **6.2** Internal State after Reset Cancellation

Table 6.1 summarizes the internal state after reset cancellation. For details on initial state of each on-chip peripheral module registers, refer to each section.

**Table 6.1** Initial Values of Register

|                         | Register   | Initial Value   |
|-------------------------|--|---|
| General register        | R0 to R14  | Undefined   |
|                         | R15 (SP)   | SP value in the vector address table  |
| Control register        | SR   | I[3:0] = 1111 (H'F), BO = 0, CS = 0,<br>reserved bits = 0, other bits = undefined |
|                         | GBR, TBR   | Undefined   |
|                         | VBR  | H'0000 0000   |
| System register         | MACH, MACL,<br>PR  | Undefined   |
|                         | PC   | PC value in the vector address table  |
| Floating point register | FPR0 to FPR15  | Undefined   |
| Floating point          | FPUL   | Undefined   |
| system register         | FPSCR  | H'00040001  |
|                         | _  | Undefined after power-on  |
|                         | Control register  System register  Floating point register | R0 to R14   |

Section 6 Reset SH72546R

## Section 7 Exception Handling

#### 7.1 Overview

#### 7.1.1 **Types of Exception Handling and Priority**

Exception handling is started by sources, such as a reset, address errors, a memory error, register bank errors, interrupts, and instructions. Table 7.1 shows their priorities. When several exception handling sources occur at once, they are processed according to the priority shown.

**Table 7.1 Types of Exception Handling and Priority** 

| Type                | Exception Handling                            | Priority |
|---------------------|---|----------|
| Reset               | Power-on reset                                | High     |
| Address             | CPU address error                             |          |
| error               | DMAC address error                            |          |
| Instruction         | FPU exception                                 |          |
|                     | Integer division exception (division by zero) |          |
|                     | Integer division exception (overflow)         | _        |
| Register bank error | Bank underflow                                |          |
|                     | Bank overflow                                 |          |
| Interrupt           | NMI   |          |
|                     | User break                                    |          |
|                     | IRQ   |          |
|                     | Memory error (RAM error/ROM error)            |          |
|                     | Software interrupt (SINT)                     | Low      |

| Туре        | <b>Exception Handling</b>                                    |   | Priority |
|-------------|--|---|----------|
| Interrupt   | On-chip peripheral modules                                   | Direct memory access controller (DMAC)  | High     |
|             |  | Compare match timer (CMT)   | _ 🛉      |
|             |  | Watchdog timer (WDT)  | _        |
|             |  | Advanced timer unit III (ATU-III)   | _        |
|             |  | A/D converter (ADC)   | _        |
|             |  | Serial communications interface (SCI)   | _        |
|             |  | Renesas serial peripheral interface (RSPI)  | _        |
|             |  | Controller area network (RCAN-TL1)  | _        |
|             |  | Automotive direct memory access controller (A-DMAC)   | _        |
| Instruction | Trap instruction (TRAPA inst                                 | ruction)  | _        |
|             | General illegal instructions (undefined code)                |   |          |
|             | branch instruction*1 (including instructions in FPU module s | fined code placed directly after a delayed g FPU instructions and FPU-related CPU tandby status), instructions that rewrite the ESBANK instruction, DIVS instruction, and | Low      |

- Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF.
  - 2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, and RTV/N.
  - 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, FMOV.S@disp12, FMOV.D@disp12, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, and MOVU.W.

#### 7.1.2 **Exception Handling Operations**

The exception handling sources are detected and exception handling starts according to the timing shown in table 7.2.

**Table 7.2** Timing of Exception Source Detection and Start of Exception Handling

| Exception            | Source                       | Timing of Source Detection and Start of Handling   |  |  |
|----------------------|------------------------------|--|--|--|
| Reset                | Power-on reset               | Starts when the $\overline{\text{RES}}$ pin changes from low to high or when the WDT overflows.  |  |  |
| Address error        |                              | Detected when instruction is decoded and starts when the   |  |  |
| Interrupts           |                              | revious executing instruction finishes executing.  |  |  |
| Register bank errors | Bank underflow               | Starts upon attempted execution of a RESBANK instruction when saving has not been performed to register banks.   |  |  |
|                      | Bank overflow                | In the state where saving has been performed to all register bank areas, starts when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.   |  |  |
| Instructions         | Trap instruction             | Starts from the execution of a TRAPA instruction.  |  |  |
|                      | General illegal instructions | Starts from the decoding of undefined code anytime except immediately after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby status).  |  |  |
|                      | Slot illegal instructions    | Starts from the decoding of undefined code placed directly after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby status), of instructions that rewrite the PC, of 32-bit instructions, of the RESBANK instruction, of the DIVS instruction, or of the DIVU instruction. |  |  |
|                      | Integer division exceptions  | Starts when detecting division-by-zero exception or overflow exception caused by division of the negative maximum value (H'80000000) by $-1$ .   |  |  |
|                      | FPU exceptions               | Starts when detecting invalid operation exception defined by IEEE standard 754, division-by-zero exception, overflow, underflow, or inexact exception.   |  |  |
|                      |                              | Also starts when qNaN or $\pm\infty$ is input to the source for a floating point operation instruction when the QIS bit in FPSCR is set.   |  |  |

When exception handling starts, the CPU operates as follows.

#### (1) Exception Handling Triggered by Reset

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception handling vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses). See section 7.1.3, Exception Handling Vector Table, for the exception handling vector table. The vector base register (VBR) is then initialized to H'00000000, the interrupt mask level bits (I3 to I0) in the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the interrupt controller (INTC) is also initialized to 0. FPSCR is also initialized to H'00040001 by a power-on reset. The program starts running from the PC address fetched from the exception handling vector table.

# (2) Exception Handling Triggered by Address Errors, Register Bank Errors, Interrupts, and Instructions

SR and PC are saved to the stack indicated by R15. In the case of interrupt exception handling other than NMI or user break with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved to the register banks. In the case of exception handling due to an address error, register bank error, NMI interrupt, user break interrupt, or instruction, saving to a register bank is not performed. When saving is performed to all register banks, automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that a register bank overflow exception is not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept a register bank overflow exception has been made (the BOVE bit in IBNR of the INTC is 1), the register bank overflow exception will be generated. In the case of interrupt exception handling, the interrupt priority level is written to the I3 to I0 bits in SR. In the case of exception handling due to an address error or instruction, the I3 to I0 bits are not affected. The start address of exception service routine is then fetched from the exception handling vector table and the program starts running from that address.

### 7.1.3 Exception Handling Vector Table

Before exception handling starts running, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception service routines. (The reset exception handling table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception handling, the start addresses of the exception service routines are fetched from the exception handling vector table, which is indicated by this vector table address.

Table 7.3 shows the vector numbers and vector table address offsets. Table 7.4 shows how vector table addresses are calculated.

**Table 7.3** Exception Handling Vector Table

| Exception Sources         |            | Vector<br>Numbers | Vector Table Address Offset |
|---------------------------|------------|-------------------|-----------------------------|
| Power-on reset            | PC         | 0                 | H'00000000 to H'00000003    |
|                           | SP         | 1                 | H'00000004 to H'00000007    |
| (Reserved for system      | ı use)     | 2                 | H'00000008 to H'0000000B    |
|                           |            | 3                 | H'0000000C to H'0000000F    |
| General illegal instru    | ction      | 4                 | H'00000010 to H'00000013    |
| (Reserved for system      | ı use)     | 5                 | H'00000014 to H'00000017    |
| Slot illegal instruction  | l          | 6                 | H'00000018 to H'0000001B    |
| (Reserved for system use) |            | 7                 | H'0000001C to H'0000001F    |
|                           |            | 8                 | H'00000020 to H'00000023    |
| CPU address error         |            | 9                 | H'00000024 to H'00000027    |
| DMAC address error        |            | 10                | H'00000028 to H'0000002B    |
| Interrupts                | NMI        | 11                | H'0000002C to H'0000002F    |
|                           | User break | 12                | H'00000030 to H'00000033    |
| FPU exception             |            | 13                | H'00000034 to H'00000037    |
| (Reserved for system use) |            | 14                | H'00000038 to H'0000003B    |
| Bank overflow             |            | 15                | H'0000003C to H'0000003F    |
| Bank underflow            |            | 16                | H'0000040 to H'0000043      |

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

Note: \* The vector numbers and vector table address offsets for each external interrupt and onchip peripheral module interrupt are given in table 8.4 in section 8, Interrupt Controller (INTC).

**Table 7.4 Calculating Exception Handling Vector Table Addresses** 

| Exception Source   | Vector Table Address Calculation  |
|--|---|
| Resets   | Vector table address = (vector table address offset)<br>= (vector number) $\times$ 4      |
| Address errors, register bank errors, interrupts, instructions | Vector table address = VBR + (vector table address offset)<br>= VBR + (vector number) × 4 |

Notes: 1. For vector table address offset, see table 7.3.

2. For vector number, see table 7.3.

#### 7.2 Resets

#### 7.2.1 Types of Reset

A reset is the highest-priority exception handling source. This LSI supports only a power-on reset. As shown in table 7.5, the CPU state, FPU state, and on-chip peripheral module registers are initialized by a power-on reset.

Table 7.5 Exception Source Detection and Exception Handling Start Timing

|          | Conditi          | Conditions for Transition to Reset State |             | Internal States                         |                 |  |  |
|----------|------------------|--|-------------|---|-----------------|--|--|
| Туре     | RES WDT Overflow |  | CPU         | On-Chip Peripheral<br>Modules, I/O Port | WTSR of WDT     |  |  |
| Power-on | Low              | _  | Initialized | Initialized                             | Initialized     |  |  |
| reset    | High             | Power-on reset                           | Initialized | Initialized                             | Not initialized |  |  |

#### 7.2.2 Power-On Reset

## (1) Power-On Reset by Means of RES Pin

When the  $\overline{RES}$  pin is driven low, this LSI enters the power-on reset state. In the power-on reset state, the internal state of the CPU and all the on-chip peripheral module registers are initialized. See section 6, Reset, for details on the power-on reset and appendix A, Pin States, for the states of individual pins during the power-on reset state.

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

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

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

#### (2) Power-On Reset Initiated by WDT

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

In this case, WTSR of the WDT is not initialized by the reset signal generated by the WDT.

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

## 7.3 Address Errors

#### 7.3.1 Address Error Sources

Address errors occur when instructions are fetched or data is read or written to, as shown in table 7.6.

Table 7.6 Bus Cycles and Address Errors

| Duc | CVA | _ |
|-----|-----|---|
| Bus | CVC | е |

| Dus Cycle           |                |  |                      |
|---------------------|----------------|--|----------------------|
| Туре                | Bus<br>Master  | Bus Cycle Description  | Address Errors       |
| Instruction C fetch | CPU            | Instruction fetched from even address                              | None (normal)        |
|                     |                | Instruction fetched from odd address                               | Address error occurs |
|                     |                | Instruction fetched from other than on-chip I/O register space     | None (normal)        |
|                     |                | Instruction fetched from on-chip I/O register space                | Address error occurs |
|                     |                | Instruction fetched from external memory space in single chip mode | Address error occurs |
| Data<br>read/write  | CPU or<br>DMAC | Word data accessed from even address                               | None (normal)        |
|                     |                | Word data accessed from odd address                                | Address error occurs |
|                     |                | Longword data accessed from a longword boundary                    | None (normal)        |
|                     |                | Longword data accessed from other than a long-word boundary        | Address error occurs |
|                     |                | Byte or word data accessed in on-chip I/O register space*1         | None (normal)        |
|                     |                | Longword data accessed in 16-bit on-chip I/O register space*1      | None (normal)        |
|                     |                | Longword data accessed in 8-bit on-chip I/O register space*1       | None (normal)        |

Puc Cyala

| bus Cycle          |                |  |                      |
|--------------------|----------------|--|----------------------|
| Туре               | Bus<br>Master  | Bus Cycle Description                              | Address Errors       |
| Data<br>read/write | CPU or<br>DMAC | External memory space accessed in single chip mode | Address error occurs |

Notes: 1. See section 5, Address Space, for details on the on-chip I/O register space.

2. If an instruction is placed within 10 bytes from the last address of the on-chip RAM space, the CPU accesses beyond the boundary in the on-chip I/O register space to fetch the instruction, causing an address error.

#### 7.3.2 Address Error Exception Handling

When an address error occurs, the bus cycle in which the address error occurred ends\*. On completion of the instruction in progress, address error exception handling starts. The CPU operates as follows:

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

Note: \* In the case of address error related to data read/write. In the case of address error related to instruction fetch, if the bus cycle in which the address error occurred does not end by the end of the PC saving to the stack in step 3, the CPU will repeat address error exception handling until the bus cycle in which the address error occurred ends.

## 7.4 Register Bank Errors

#### 7.4.1 Register Bank Error Sources

#### (1) Bank Overflow

In the state where saving has already been performed to all register bank areas, bank overflow occurs when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is set to 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.

#### (2) Bank Underflow

Bank underflow occurs when an attempt is made to execute a RESBANK instruction while saving has not been performed to register banks.

## 7.4.2 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. The CPU operates as follows:

- 1. The start address of exception service routine which corresponds to the register bank error that occurred is fetched from the exception handling vector table.
- 2. The status register (SR) is saved to the stack.
- 3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction for a bank overflow, and the start address of the executed RESBANK instruction for a bank underflow.
  - To prevent multiple interrupts from occurring at a bank overflow, the interrupt priority level that caused the bank overflow is written to the interrupt mask level bits (I3 to I0) in the status register (SR).
- 4. After jumping to the start address of exception service routine fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

## 7.5 Interrupts

## 7.5.1 Interrupt Sources

Table 7.7 shows the sources that start up interrupt exception handling. These are divided into NMI, user breaks, IRQ, SINT, and on-chip peripheral modules.

**Table 7.7 Interrupt Sources** 

| Туре                      | Request Source                                      | Number of<br>Sources |
|---------------------------|---|----------------------|
| NMI                       | NMI pin (external input)                            | 1                    |
| User break                | User break controller (UBC)                         | 1                    |
| IRQ                       | IRQ0 to IRQ7 pins (external input)                  | 8                    |
| Memory error              | RAM/ROM   | 2                    |
| SINT                      | Software interrupt                                  | 15                   |
| On-chip peripheral module | Direct memory access controller (DMAC)              | 16                   |
|                           | Compare match timer (CMT)                           | 2                    |
|                           | Watchdog timer (WDT)                                | 1                    |
|                           | Advanced timer unit III (ATU-III)                   | 153                  |
|                           | A/D converter (ADC)                                 | 26                   |
|                           | Serial communications interface (SCI)               | 20                   |
|                           | Renesas serial peripheral interface (RSPI)          | 9                    |
|                           | Controller area network (RCAN-TL1)                  | 18                   |
|                           | Automotive direct memory access controller (A-DMAC) | 1                    |

Each interrupt source is allocated a different vector number and vector table offset. See table 8.4 in section 8, Interrupt Controller (INTC), for more information on vector numbers and vector table address offsets.

## 7.5.2 Interrupt Priority Level

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

The priority of interrupts is expressed as priority levels 0 to 16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, thus it is always accepted. Priority levels of the user break interrupt and the memory error interrupt are 15. The priority levels of software interrupts (SINT) are fixed to priority 15 to 1 for each source of SINT15 to SINT1. For example, the SINT15 priority level is 15, and the SINT14 priority level is 14. Priority levels of IRQ interrupts and on-chip peripheral module interrupts can be set freely using the interrupt priority registers 01 to 29 (IPR01 to IPR29) of the INTC as shown in table 7.8. The priority levels that can be set are 0 to 15. Level 16 cannot be set. See section 8.3.1, Interrupt Priority Registers 01 to 29 (IPR01 to IPR29), for details on IPR01 to IPR29.

**Table 7.8** Interrupt Priority

| Туре                      | Priority Level | Comment                                     |
|---------------------------|----------------|---|
| NMI                       | 16             | Fixed priority level. Cannot be masked.     |
| User break                | 15             | Fixed priority level.                       |
| IRQ                       | 0 to 15        | Set with interrupt priority register (IPR). |
| Memory error              | 15             | Fixed priority level.                       |
| SINT15 to SINT1           | 15 to 1        | Fixed priority level.                       |
| On-chip peripheral module | 0 to 15        | Set with interrupt priority register (IPR). |

## 7.5.3 Interrupt Exception Handling

When an interrupt occurs, its priority level is ascertained by the interrupt controller (INTC). NMI is always accepted, but other interrupts are only accepted if they have a priority level higher than the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, interrupt exception handling starts. In interrupt exception handling, the CPU fetches the exception service routine start address which corresponds to the accepted interrupt from the exception handling vector table, and saves SR and the program counter (PC) to the stack. In the case of interrupt exception handling other than NMI or user break with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved in the register banks. In the case of exception handling due to an address error, NMI interrupt, user break interrupt, or instruction, saving is not performed to the register banks. If saying has been performed to all register banks (0 to 14), automatic saying to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that a register bank overflow exception is not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept a register bank overflow exception has been made (the BOVE bit in IBNR of the INTC is 1), the register bank overflow exception occurs. Next, the priority level value of the accepted interrupt is written to the I3 to I0 bits in SR. For NMI, however, the priority level is 16, but the value set in the I3 to I0 bits is H'F (level 15). Then, after jumping to the start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch. See section 8.6, Operation, for further details on interrupt exception handling.

# 7.6 Exceptions Triggered by Instructions

## 7.6.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by trap instructions, slot illegal instructions, general illegal instructions, integer division exceptions, and FPU exceptions as shown in table 7.9.

**Table 7.9** Types of Exceptions Triggered by Instructions

| Туре                        | Source Instruction  | Comment   |  |  |  |  |
|-----------------------------|---|---|--|--|--|--|
| Trap instruction            | TRAPA   |   |  |  |  |  |
| Slot illegal instruction    | Undefined code placed immediately after a delayed branch instruction (delay slot)   | Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF   |  |  |  |  |
|                             | (including FPU instructions and FPU-related CPU instructions in FPU module standby state), instructions that rewrite the PC, 32-bit instructions, RESBANK | Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, and RTV/N   |  |  |  |  |
|                             | instruction, DIVS instruction, and DIVU instruction   | 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, FMOV.S@disp12, FMOV.D@disp12, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W. |  |  |  |  |
| General illegal instruction | Undefined code anywhere besides in a delay slot (including FPU instructions and FPU-related CPU instructions in FPU module standby status)                |   |  |  |  |  |
| Integer division exception  | Division by zero  | DIVU, DIVS  |  |  |  |  |
|                             | Negative maximum value ÷ (-1)   | DIVS  |  |  |  |  |
| FPU exceptions              | Starts when detecting invalid operation exception defined by IEEE754, division-by-zero exception, overflow, underflow, or inexact exception.              | FADD, FSUB, FMUL, FDIV, FMAC,<br>FCMP/EQ, FCMP/GT, FNEG, FABS,<br>FLOAT, FTRC, FCNVDS, FCNVSD,<br>FSQRT   |  |  |  |  |

### 7.6.2 Trap Instructions

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

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

## 7.6.3 Slot Illegal Exceptions

An instruction placed immediately after a delayed branch instruction is said to be placed in a delay slot. When the instruction placed in the delay slot is undefined code (including FPU instructions and FPU-related CPU instructions in FPU module standby state), an instruction that rewrites the PC, a 32-bit instruction, an RESBANK instruction, a DIVS instruction, or a DIVU instruction, slot illegal exception handling starts when such kind of instruction is decoded. The CPU operates as follows:

- 1. The start address of exception service routine is fetched from the exception handling vector table.
- 2. The status register (SR) is saved to the stack.
- 3. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code (including FPU instructions and FPU-related CPU instructions in FPU module standby state), the instruction that rewrites the PC, the 32-bit instruction, the RESBANK instruction, the DIVS instruction, or the DIVU instruction.
- 4. After jumping to the start address of exception service routine fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

## 7.6.4 General Illegal Instructions

When undefined code placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state) is decoded, general illegal instruction exception handling starts. The CPU handles general illegal instructions in the same way as slot illegal instructions. Unlike processing of slot illegal instructions, however, the program counter value stored is the start address of the undefined code.

### 7.6.5 Integer Division Exceptions

When an integer division instruction performs division by zero or the result of integer division overflows, integer division instruction exception handling starts. The instructions that may become the source of division-by-zero exception are DIVU and DIVS. The only source instruction of overflow exception is DIVS, and overflow exception occurs only when the negative maximum value is divided by -1. The CPU operates as follows:

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

### 7.6.6 FPU Exceptions

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

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

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

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

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

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

# 7.7 When Exception Sources Are Not Accepted

When an address error, FPU exception, register bank error (overflow), or interrupt is generated immediately after a delayed branch instruction, it is sometimes not accepted immediately but stored instead, as shown in table 7.10. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 7.10 Exception Source Generation Immediately after Delayed Branch Instruction

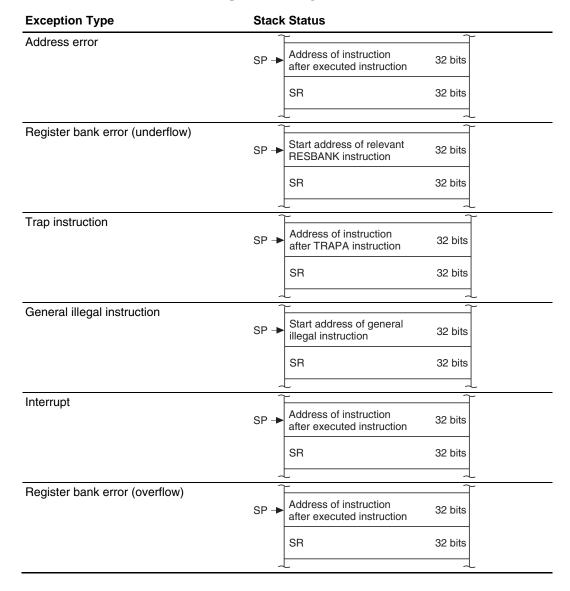
|   | Exception Source |               |                                   |              |  |  |  |  |  |
|---|------------------|---------------|-----------------------------------|--------------|--|--|--|--|--|
| Point of Occurrence Address Erro                |                  | FPU Exception | Register Bank<br>Error (Overflow) | Interrupt    |  |  |  |  |  |
| Immediately after a delayed branch instruction* | Not accepted     | Not accepted  | Not accepted                      | Not accepted |  |  |  |  |  |

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

# 7.8 Stack Status after Exception Handling Ends

The state of the stack after exception handling ends is as shown in table 7.11.

Table 7.11 Stack Status After Exception Handling Ends



| Exception Type               | Stack Status |  |         |  |  |  |  |
|------------------------------|--------------|--|---------|--|--|--|--|
| FPU exception                | SP →         | Address of instruction after executed instruction      | 32 bits |  |  |  |  |
|                              |              | SR   | 32 bits |  |  |  |  |
|                              |              | L  |         |  |  |  |  |
| Slot illegal instruction     | SP →         | Jump destination address of delayed branch instruction | 32 bits |  |  |  |  |
|                              |              | SR   | 32 bits |  |  |  |  |
|                              |              |  |         |  |  |  |  |
| Integer division instruction | SP →         | Start address of relevant integer division instruction | 32 bits |  |  |  |  |
|                              |              | SR   | 32 bits |  |  |  |  |
|                              | _            |  |         |  |  |  |  |

# 7.9 Usage Notes

## 7.9.1 Value of Stack Pointer (SP)

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

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

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

## 7.9.3 Address Errors Caused by Stacking of Address Error Exception Handling

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception handling (interrupts, etc.) and address error exception handling will start up as soon as the first exception handling is ended. Address errors will then also occur in the stacking for this address error exception handling. To ensure that address error exception handling does not go into an endless loop, no address errors are accepted at that point. This allows program control to be shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle (write) is executed. During stacking of the status register (SR) and program counter (PC), the SP is decremented by 4 for both, so the value of SP will not be a multiple of four after the stacking either. The address value output during stacking is the SP value, so the address where the error occurred is itself output. This means the write data stacked will be undefined.

# Section 8 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The priority of each interrupt can be set by the INTC registers, and interrupts are processed according to the user-set priority.

### 8.1 Features

- 16 levels of interrupt priority can be set
   By setting the twenty-nine interrupt priority registers, the priorities of IRQ interrupts and onchip peripheral module interrupts can be selected from 16 levels for request sources.
- NMI noise canceller function
   An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception handler, the pin state can be checked, enabling it to be used as the noise canceller function.
- Register banks
   This LSI has register banks that enable register saving and restoring required in the interrupt processing to be performed at high speed.
- Software interrupt (SINT)
   By setting the software interrupt register, an interrupt with a given priority can be generated from a program.

Figure 8.1 shows a block diagram of the INTC.

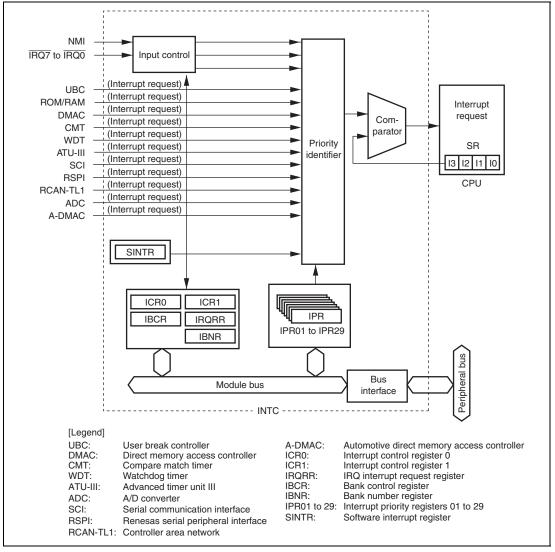


Figure 8.1 Block Diagram of INTC

# 8.2 Input/Output Pins

Table 8.1 shows the pin configuration of the INTC.

**Table 8.1** Pin Configuration

| Pin Name                        | Symbol       | I/O   | Function   |
|---------------------------------|--------------|-------|--|
| Nonmaskable interrupt input pin | NMI          | Input | Input pin for the nonmaskable interrupt request signal |
| Interrupt request input pins    | IRQ7 to IRQ0 | Input | Input pins for maskable interrupt request signals      |

# **8.3** Register Descriptions

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

**Table 8.2** Register Configuration

| Register Name                  | Symbol | R/W     | Initial Value | Address    | Access<br>Size |
|--------------------------------|--------|---------|---------------|------------|----------------|
| Interrupt control register 0   | ICR0   | R/W     | *1            | H'FFFE0800 | 16, 32         |
| Interrupt control register 1   | ICR1   | R/W     | H'0000        | H'FFFE0802 | 16, 32         |
| IRQ interrupt request register | IRQRR  | R/(W)*2 | H'0000        | H'FFFE0806 | 8, 16, 32      |
| Bank control register          | IBCR   | R/W     | H'0000        | H'FFFE080C | 16, 32         |
| Bank number register           | IBNR   | R/W     | H'0000        | H'FFFE080E | 16, 32         |
| Software interrupt register 1  | SINTR1 | R/W     | H'00          | H'FFFE0810 | 8, 16, 32      |
| Software interrupt register 2  | SINTR2 | R/W     | H'00          | H'FFFE0811 | 8, 16, 32      |
| Software interrupt register 3  | SINTR3 | R/W     | H'00          | H'FFFE0812 | 8, 16, 32      |
| Software interrupt register 4  | SINTR4 | R/W     | H'00          | H'FFFE0813 | 8, 16, 32      |
| Software interrupt register 5  | SINTR5 | R/W     | H'00          | H'FFFE0814 | 8, 16, 32      |
| Software interrupt register 6  | SINTR6 | R/W     | H'00          | H'FFFE0815 | 8, 16, 32      |
| Software interrupt register 7  | SINTR7 | R/W     | H'00          | H'FFFE0816 | 8, 16, 32      |
| Software interrupt register 8  | SINTR8 | R/W     | H'00          | H'FFFE0817 | 8, 16, 32      |
| Interrupt priority register 01 | IPR01  | R/W     | H'0000        | H'FFFE0818 | 16, 32         |

| Register Name                  | Symbol  | R/W | Initial Value | Address    | Access<br>Size |
|--------------------------------|---------|-----|---------------|------------|----------------|
| Interrupt priority register 02 | IPR02   | R/W | H'0000        | H'FFFE081A | 16, 32         |
| Software interrupt register 9  | SINTR9  | R/W | H'00          | H'FFFE0828 | 8, 16, 32      |
| Software interrupt register 10 | SINTR10 | R/W | H'00          | H'FFFE0829 | 8, 16, 32      |
| Software interrupt register 11 | SINTR11 | R/W | H'00          | H'FFFE082A | 8, 16, 32      |
| Software interrupt register 12 | SINTR12 | R/W | H'00          | H'FFFE082B | 8, 16, 32      |
| Software interrupt register 13 | SINTR13 | R/W | H'00          | H'FFFE082C | 8, 16, 32      |
| Software interrupt register 14 | SINTR14 | R/W | H'00          | H'FFFE082D | 8, 16, 32      |
| Software interrupt register 15 | SINTR15 | R/W | H'00          | H'FFFE082E | 8, 16, 32      |
| Interrupt priority register 03 | IPR03   | R/W | H'0000        | H'FFFE0C00 | 16, 32         |
| Interrupt priority register 04 | IPR04   | R/W | H'0000        | H'FFFE0C02 | 16, 32         |
| Interrupt priority register 05 | IPR05   | R/W | H'0000        | H'FFFE0C04 | 16, 32         |
| Interrupt priority register 06 | IPR06   | R/W | H'0000        | H'FFFE0C06 | 16, 32         |
| Interrupt priority register 07 | IPR07   | R/W | H'0000        | H'FFFE0C08 | 16, 32         |
| Interrupt priority register 08 | IPR08   | R/W | H'0000        | H'FFFE0C0A | 16, 32         |
| Interrupt priority register 09 | IPR09   | R/W | H'0000        | H'FFFE0C0C | 16, 32         |
| Interrupt priority register 10 | IPR10   | R/W | H'0000        | H'FFFE0C0E | 16, 32         |
| Interrupt priority register 11 | IPR11   | R/W | H'0000        | H'FFFE0C10 | 16, 32         |
| Interrupt priority register 12 | IPR12   | R/W | H'0000        | H'FFFE0C12 | 16, 32         |
| Interrupt priority register 13 | IPR13   | R/W | H'0000        | H'FFFE0C14 | 16, 32         |
| Interrupt priority register 14 | IPR14   | R/W | H'0000        | H'FFFE0C16 | 16, 32         |
| Interrupt priority register 15 | IPR15   | R/W | H'0000        | H'FFFE0C18 | 16, 32         |
| Interrupt priority register 16 | IPR16   | R/W | H'0000        | H'FFFE0C1A | 16, 32         |
| Interrupt priority register 17 | IPR17   | R/W | H'0000        | H'FFFE0C1C | 16, 32         |
| Interrupt priority register 18 | IPR18   | R/W | H'0000        | H'FFFE0C1E | 16, 32         |
| Interrupt priority register 19 | IPR19   | R/W | H'0000        | H'FFFE0C20 | 16, 32         |
| Interrupt priority register 20 | IPR20   | R/W | H'0000        | H'FFFE0C22 | 16, 32         |
| Interrupt priority register 21 | IPR21   | R/W | H'0000        | H'FFFE0C24 | 16, 32         |
| Interrupt priority register 22 | IPR22   | R/W | H'0000        | H'FFFE0C26 | 16, 32         |
| Interrupt priority register 23 | IPR23   | R/W | H'0000        | H'FFFE0C28 | 16, 32         |
| Interrupt priority register 24 | IPR24   | R/W | H'0000        | H'FFFE0C2A | 16, 32         |

Page 126 of 1824

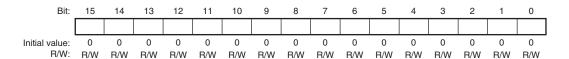
| Register Name                  | Symbol | R/W | Initial Value | Address    | Access<br>Size |
|--------------------------------|--------|-----|---------------|------------|----------------|
| Interrupt priority register 25 | IPR25  | R/W | H'0000        | H'FFFE0C2C | 16, 32         |
| Interrupt priority register 26 | IPR26  | R/W | H'0000        | H'FFFE0C2E | 16, 32         |
| Interrupt priority register 27 | IPR27  | R/W | H'0000        | H'FFFE0C30 | 16, 32         |
| Interrupt priority register 28 | IPR28  | R/W | H'0000        | H'FFFE0C32 | 16, 32         |
| Interrupt priority register 29 | IPR29  | R/W | H'0000        | H'FFFE0C34 | 16, 32         |

Notes: 1. H'8000 when the NMI pin is high and H'0000 when low.

2. Only 0 can be written after reading 1, to clear the flag.

#### **Interrupt Priority Registers 01 to 29 (IPR01 to IPR29)** 8.3.1

IPR01 to IPR29 are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for IRQ interrupts and on-chip peripheral module interrupts. Table 8.3 shows the correspondence between the interrupt request sources and the bits in IPR01 to IPR29.



**Interrupt Request Sources and IPR01 to IPR29** Table 8.3

| Register Name                  | Bits 15 to 12           | Bits 11 to 8            | Bits 7 to 4             | Bits 3 to 0     |
|--------------------------------|-------------------------|-------------------------|-------------------------|-----------------|
| Interrupt priority register 01 | IRQ0                    | IRQ1                    | IRQ2                    | IRQ3            |
| Interrupt priority register 02 | IRQ4                    | IRQ5                    | IRQ6                    | IRQ7            |
| Interrupt priority register 03 | DMAC0                   | DMAC1                   | DMAC2                   | DMAC3           |
| Interrupt priority register 04 | DMAC4                   | DMAC5                   | DMAC6                   | DMAC7           |
| Interrupt priority register 05 | СМТО                    | CMT1                    | Reserved                | WDT             |
| Interrupt priority register 06 | ATU-A<br>(ICIA0, ICIA1) | ATU-A<br>(ICIA2, ICIA3) | ATU-A<br>(ICIA4, ICIA5) | ATU-A<br>(OVIA) |

| Register Name                     | Bits 15 to 12                   | Bits 11 to 8                    | Bits 7 to 4                     | Bits 3 to 0                     |
|-----------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Interrupt priority<br>register 07 | ATU-B<br>(CMIB0, CMIB1)         | ATU-B<br>(CMIB6, ICIB0)         | ATU-C0<br>(IMIC00 to<br>IMIC03) | ATU-C0<br>(OVIC0)               |
| Interrupt priority register 08    | ATU-C1<br>(IMIC10 to<br>IMIC13) | ATU-C1<br>(OVIC1)               | ATU-C2<br>(IMIC20 to<br>IMIC23) | ATU-C2<br>(OVIC2)               |
| Interrupt priority<br>register 09 | ATU-C3<br>(IMIC30 to<br>IMIC33) | ATU-C3<br>(OVIC3)               | ATU-C4<br>(IMIC40 to<br>IMIC43) | ATU-C4<br>(OVIC4)               |
| Interrupt priority register 10    | ATU-D0<br>(CMID00 to<br>CMID03) | ATU-D0<br>(OVI1D0, OVI2D0)      | ATU-D0<br>(UDID00 to<br>UDID03) | ATU-D1<br>(CMID10 to<br>CMID13) |
| Interrupt priority register 11    | ATU-D1<br>(OVI1D1, OVI2D1)      | ATU-D1<br>(UDID10 to<br>UDID13) | ATU-D2<br>(CMID20 to<br>CMID23) | ATU-D2<br>(OVI1D2,<br>OVI2D2)   |
| Interrupt priority register 12    | ATU-D2<br>(UDID20 to<br>UDID23) | ATU-D3<br>(CMID30 to<br>CMID33) | ATU-D3<br>(OVI1D3, OVI2D3)      | ATU-D3<br>(UDID30 to<br>UDID33) |
| Interrupt priority register 13    | Reserved                        | Reserved                        | Reserved                        | Reserved                        |
| Interrupt priority register 14    | Reserved                        | Reserved                        | ATU-E0<br>(CMIE00 to<br>CMIE03) | ATU-E1<br>(CMIE10 to<br>CMIE13) |
| Interrupt priority register 15    | ATU-E2<br>(CMIE20 to<br>CMIE23) | ATU-E3<br>(CMIE30 to<br>CMIE33) | ATU-E4<br>(CMIE40 to<br>CMIE43) | ATU-E5<br>(CMIE50 to<br>CMIE53) |
| Interrupt priority register 16    | ATU-F<br>(ICIF0 to ICIF3)       | ATU-F<br>(ICIF4 to ICIF7)       | ATU-F<br>(ICIF8 to ICIF11)      | ATU-F<br>(ICIF12 to ICIF15)     |
| Interrupt priority register 17    | ATU-F<br>(ICIF16 to ICIF19)     | Reserved                        | Reserved                        | Reserved                        |
| Interrupt priority register 18    | ATU-F<br>(OVIF0 to OVIF3)       | ATU-F<br>(OVIF4 to OVIF7)       | ATU-F<br>(OVIF8 to OVIF11)      | ATU-F<br>(OVIF12 to<br>OVIF15)  |
| Interrupt priority<br>register 19 | ATU-F<br>(OVIF16 to<br>OVIF19)  | Reserved                        | Reserved                        | Reserved                        |
| Interrupt priority register 20    | ATU-G<br>(CMIG0 to CMIG3)       | ATU-G<br>(CMIG4, CMIG5)         | ATU-H<br>(CMIH)                 | Reserved                        |
| Interrupt priority register 21    | ATU-J<br>(DFIJ0, DFIJ1)         | ATU-J<br>(OVIJ0, OVIJ1)         | ATU-J<br>(DOVIJ0, DOVIJ1)       | Reserved                        |

| Register Name                  | Bits 15 to 12               | Bits 11 to 8                 | Bits 7 to 4             | Bits 3 to 0             |
|--------------------------------|-----------------------------|------------------------------|-------------------------|-------------------------|
| Interrupt priority register 22 | ADC<br>(ADI0)               | ADC<br>(ADI1)                | ADC<br>(ADID0 to ADID3) | ADC<br>(ADID4 to ADID7) |
| Interrupt priority register 23 | ADC<br>(ADID8 to<br>ADID11) | ADC<br>(ADID12 to<br>ADID15) | ADC<br>(ADID40)         | ADC<br>(ADID41)         |
| Interrupt priority register 24 | ADC<br>(ADID42)             | ADC<br>(ADID43)              | ADC<br>(ADID44)         | ADC<br>(ADID45)         |
| Interrupt priority register 25 | ADC<br>(ADID46)             | ADC<br>(ADID47)              | Reserved                | Reserved                |
| Interrupt priority register 26 | SCI_A                       | SCI_B                        | SCI_C                   | SCI_D                   |
| Interrupt priority register 27 | SCI_E                       | RSPI_A                       | RSPI_B                  | RSPI_C                  |
| Interrupt priority register 28 | RCAN_A                      | RCAN_B                       | RCAN_C                  | Reserved                |
| Interrupt priority register 29 | A-DMAC                      | Reserved                     | Reserved                | Reserved                |

As shown in table 8.3, by setting the 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) with values from H'0 (0000) to H'F (1111), the priority of each corresponding interrupt is set. Setting of H'0 means priority level 0 (the lowest level) and H'F means priority level 15 (the highest level).

IPR01 to IPR29 are initialized to H'0000 by a power-on reset.

## 8.3.2 Interrupt Control Register 0 (ICR0)

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

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

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

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

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

#### 8.3.3 **Interrupt Control Register 1 (ICR1)**

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

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

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |
|-----|----------|------------------|--------|---|
| —   | Dit Name | value            | IT/ WV | Description   |
| 15  | IRQ71S   | 0                | R/W    | IRQ Sense Select  |
| 14  | IRQ70S   | 0                | R/W    | These bits select whether interrupt signals   |
| 13  | IRQ61S   | 0                | R/W    | corresponding to pins IRQ7 to IRQ0 are detected by a low level, falling edge, rising edge, or both edges. |
| 12  | IRQ60S   | 0                | R/W    | 00: Interrupt request is detected on low level of IRQn  |
| 11  | IRQ51S   | 0                | R/W    | input   |
| 10  | IRQ50S   | 0                | R/W    | 01: Interrupt request is detected on falling edge of  |
| 9   | IRQ41S   | 0                | R/W    | TRQn input  — 10: Interrupt request is detected on rising edge of   |
| 8   | IRQ40S   | 0                | R/W    | To. Interrupt request is detected on hising edge of IRQn input  |
| 7   | IRQ31S   | 0                | R/W    | 11: Interrupt request is detected on both edges of  |
| 6   | IRQ30S   | 0                | R/W    | IRQn input  |
| 5   | IRQ21S   | 0                | R/W    | _   |
| 4   | IRQ20S   | 0                | R/W    | _   |
| 3   | IRQ11S   | 0                | R/W    |   |
| 2   | IRQ10S   | 0                | R/W    | _   |
| 1   | IRQ01S   | 0                | R/W    |   |
| 0   | IRQ00S   | 0                | R/W    |   |

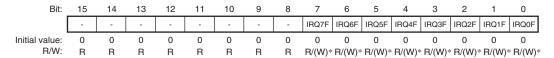
[Legend]

n = 7 to 0

## 8.3.4 IRQ Interrupt Request Register (IRQRR)

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

IRQRR is initialized by a power-on reset.



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

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 15 to 8 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial<br>Value | R/W    | Description  |
|-----|----------|------------------|--------|--|
| 7   | IRQ7F    | 0                | R/(W)* | IRQ Interrupt Request  |
| 6   | IRQ6F    | 0                | R/(W)* | These bits indicate the status of the IRQ7 to IRQ0 – interrupt requests. |
| 5   | IRQ5F    | 0                | R/(W)* | Level detection:   |
| 4   | IRQ4F    | 0                | R/(W)* | 0: IRQn interrupt request has not occurred                               |
| 3   | IRQ3F    | 0                | R/(W)* | [Clearing condition]   |
| 2   | IRQ2F    | 0                | R/(W)* | IRQn input is high   |
| 1   | IRQ1F    | 0                | R/(W)* | 1: IRQn interrupt has occurred   |
| 0   | IRQ0F    | 0                | R/(W)* | <ul><li>[Setting condition]</li><li>IRQn input is low</li></ul>          |
|     |          |                  |        | Edge detection:  |
|     |          |                  |        | 0: IRQn interrupt request is not detected                                |

[Clearing conditions]

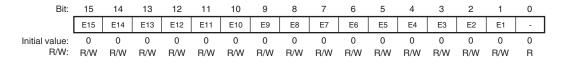
- Cleared by reading IRQnF while IRQnF = 1, then writing 0 to IRQnF
- Cleared by executing IRQn interrupt exception handling
- 1: IRQn interrupt request is detected [Setting condition]
- Edge corresponding to IRQn1S or IRQn0S of ICR1 has occurred at IRQn pin

[Legend]

n = 7 to 0

## 8.3.5 Bank Control Register (IBCR)

IBCR is a 16-bit register that enables or disables use of register banks for each interrupt priority level. IBCR is initialized to H'0000 by a power-on reset.



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

## 8.3.6 Bank Number Register (IBNR)

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

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

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

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

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

#### 8.3.7 **Software Interrupt Registers 1 to 15 (SINTR1 to SINTR15)**

SINTR1 to SINTR15 are 8-bit registers that control software interrupts 1 to 15 (SINT1 to SINT15). Writing H'01 to this register generates the software interrupts 1 to 15 (SINT1 to SINT15). Writing H'00 while handling the generated interrupts clears the interrupt sources. When SINTR1 to SINTR15 are read, the current register value is read.

SINTR1 to SINTR15 are initialized to H'00 by a power-on reset.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
|----------------|---|---|---|---|---|---|---|-------|
|                | - | - | - | - | - | - | - | SINTC |
| 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<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 1 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 0      | SINTC    | 0                | R/W | Software Interrupt Request   |
|        |          |                  |     | These bits generate software interrupts 1 to 15 (SINT1 to SINT15).   |
|        |          |                  |     | [Read operation]   |
|        |          |                  |     | Current bit value is read.   |
|        |          |                  |     | [Write operation]  |
|        |          |                  |     | Writing 1: Generates interrupts                                      |
|        |          |                  |     | Writing 1 to this bit is prohibited when the SINTC bit is 1          |
|        |          |                  |     | Writing 0: Clears interrupt sources                                  |

## 8.4 Interrupt Sources

There are six types of interrupt sources: the NMI interrupt, user break, IRQ, memory error interrupt, software interrupts (SINT), and interrupts from on-chip peripheral modules. Each interrupt has a priority level (0 to 16). A level of 0 corresponds to the lowest and a level of 16 corresponds to the highest. When the level is set to 0, the interrupt is masked at all times.

## 8.4.1 NMI Interrupt

The NMI interrupt has a priority level of 16 and is received at all times. The edge of the NMI signal is detected as an NMI interrupt and the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) selects whether the rising edge or falling edge is detected.

Although the priority level of the NMI interrupt is 16, the interrupt mask level bits (I3 to I0) in the status register (SR) are set to level 15 in the NMI interrupt exception handler.

## 8.4.2 User Break Interrupt

A user break interrupt occurs when a break condition set in the user break controller (UBC) is satisfied and has a priority level of 15. Bits I3 to I0 in SR is set to level 15 in the user break interrupt exception handler. For details on the user break interrupt, see section 9, User Break Controller (UBC).

# 8.4.3 IRQ Interrupts

An IRQ interrupt is input on pins  $\overline{IRQ7}$  to  $\overline{IRQ0}$ . The low level, falling edge, rising edge, or both edge of the IRQ signals is detected and the edge to be detected can be selected individually for each pin by the IRQ sense select bits (IRQ71S to IRQ01S and IRQ70S to IRQ00S) in interrupt control register 1 (ICR1). The priority level can be set individually in a range from 0 to 15 for each pin by interrupt priority registers 01 and 02 (IPR01 and IPR02).

When using the low-level sensing for IRQ interrupts, an interrupt is requested to the INTC while signals  $\overline{IRQ7}$  to  $\overline{IRQ0}$  are driven low. When the signals are driven high, the interrupt stops to be requested. Whether or not an IRQ interrupt has occurred can be checked by reading the IRQ interrupt request bits (IRQ7R to IRQ0R) in the IRQ interrupt request register (IRQRR).

When using a edge sensing for IRQ interrupts, an interrupt is requested to the INTC when an interrupt request is detected due to changes in signals  $\overline{IRQ7}$  to  $\overline{IRQ0}$ . The interrupt request is held until the interrupt is received and whether or not an interrupt has requested can be checked by reading bits IRQ7R to IRQ0R in IRQRR. The request can be cleared by writing 0 to these bits after reading them as 1.

In the IRQ interrupt exception handler, bits I3 to I0 in SR are set to the priority level of the received IRQ interrupt.

Before execution of the RTE instruction to exit the IRQ interrupt exception handler, confirm that the interrupt has been cleared by reading the IRQ interrupt request register (IRQRR). Otherwise, the interrupt is requested again erroneously.

#### 8.4.4 **Memory Error Interrupt**

For details on the sources generating a memory error, see section 25, ROM, and section 28, RAM.

#### 8.4.5 **Software Interrupts (SINT)**

A software interrupt (SINT) is generated by setting SINTR1 to SINTR15 by software. The interrupt priority levels of the software interrupts (SINT) are fixed as shown in table 8.4. In the SINT interrupt exception handles, bits I3 to I0 in SR is set to the priority level of the received software interrupt (SINT).

#### 8.4.6 **On-Chip Peripheral Module Interrupts**

The following on-chip peripheral modules can generate on-chip peripheral module interrupts.

- Direct memory access controller (DMAC)
- Compare match timer (CMT)
- Watchdog timer (WDT)
- Advanced timer unit III (ATU-III)
- A/D converter (ADC)
- Serial communications interface (SCI)
- Renesas serial peripheral interface (RSPI)
- Controller area network (RCAN-TL1)
- Automotive direct memory access controller (A-DMAC)

Since each source is assigned to a unique interrupt vector, the source does not need to be identified in the interrupt exception handler. A priority level in a range from 0 to 15 can be set for each module by interrupt priority registers 03 to 29 (IPR03 to IPR29). In the exception handler for the on-chip peripheral module interrupt, bits I3 to I0 in SR is set to the priority level of the received on-chip peripheral module interrupt.

# 8.5 Interrupt Exception Handling Vector Table and Priority

Table 8.4 lists interrupt sources and their vector numbers, vector table address offsets, and interrupt priority levels.

Each interrupt source is assigned to a unique vector number and a unique vector table address offset. Vector table addresses are calculated from the vector numbers and vector table address offsets. In the interrupt exception handler, the start address is fetched from the vector table pointed to by the vector table address. For details of calculation of the vector table address, see table 7.4, Calculating Exception Handling Vector Table Addresses, in section 7, Exception Handling.

The priorities of IRQ interrupts and on-chip peripheral module interrupts can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers 01 to 29 (IPR01 to IPR29). However, if two or more interrupts specified by the same IPR among IPR03 to IPR29 occur, the interrupts are processed according to the priority levels defined as shown in the Priority within IPR Setting Unit columns in table 8.4, and the priority levels cannot be changed. The priority levels of IRQ and on-chip peripheral module interrupts are initialized to 0 by a power-on reset. If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, they are processed according to the default priority levels shown in the Default Priority columns in table 8.4.

Table 8.4 Interrupt Exception Handling Vectors and Priority Levels

|             |                 | Inte   | errupt Vector                  |                                    |                         | Priority                      |                     |
|-------------|-----------------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interrup    | t Source Number | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| NMI         |                 | 11     | H'0000002C to<br>H'0000002F    | 16                                 | _                       | _                             | High<br><b>∱</b>    |
| User bre    | ak              | 12     | H'00000030 to<br>H'00000033    | 15                                 | _                       | _                             | -                   |
| IRQ         | IRQ0            | 64     | H'00000100 to<br>H'00000103    | 0 to 15 (0)                        | IPR01 (15 to 12)        | _                             | -                   |
|             | IRQ1            | 65     | H'00000104 to<br>H'00000107    | 0 to 15 (0)                        | IPR01 (11 to 8)         | _                             | -                   |
|             | IRQ2            | 66     | H'00000108 to<br>H'0000010B    | 0 to 15 (0)                        | IPR01 (7 to 4)          | _                             | -                   |
|             | IRQ3            | 67     | H'0000010C to<br>H'0000010F    | 0 to 15 (0)                        | IPR01 (3 to 0)          | _                             | -                   |
|             | IRQ4            | 68     | H'00000110 to<br>H'00000113    | 0 to 15 (0)                        | IPR02 (15 to 12)        | _                             | -                   |
|             | IRQ5            | 69     | H'00000114 to<br>H'00000117    | 0 to 15 (0)                        | IPR02 (11 to 8)         | _                             |                     |
|             | IRQ6            | 70     | H'00000118 to<br>H'0000011B    | 0 to 15 (0)                        | IPR02 (7 to 4)          | _                             |                     |
|             | IRQ7            | 71     | H'0000011C to<br>H'0000011F    | 0 to 15 (0)                        | IPR02 (3 to 0)          | _                             |                     |
| RAM/<br>ROM | RAME            | 79     | H'0000013C to<br>H'0000013F    | 15                                 | _                       | _                             | -                   |
|             | FIFE            | 82     | H'00000148 to<br>H'0000014B    | 15                                 | _                       | _                             |                     |
| SINT        | SINT15          | 93     | H'00000174 to<br>H'00000177    | 15                                 | _                       | _                             | -                   |
|             | SINT14          | 94     | H'00000178 to<br>H'0000017B    | 14                                 | _                       | _                             |                     |
|             | SINT13 95       |        | H'0000017C to<br>H'0000017F    | 13                                 | _                       | _                             |                     |
|             | SINT12          | 96     | H'00000180 to<br>H'00000183    | 12                                 |                         | _                             | Low                 |

|                         |        |        | Interrupt Vector               |                                    |              |                  | Priority<br>within IPR |                  |
|-------------------------|--------|--------|--------------------------------|------------------------------------|--------------|------------------|------------------------|------------------|
| Interrupt Source Number |        | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | •            |                  | Default<br>Priority    |                  |
| SINT                    | SINT11 |        | 97                             | H'00000184 to<br>H'00000187        | 11           | _                |                        | High<br><b>∱</b> |
|                         | SINT10 |        | 98                             | H'00000188 to<br>H'0000018B        | 10           | _                | _                      | _                |
|                         | SINT9  |        | 99                             | H'0000018C to<br>H'0000018F        | 9            | _                | _                      | _                |
|                         | SINT8  |        | 100                            | H'00000190 to<br>H'00000193        | 8            | _                | _                      | _                |
|                         | SINT7  |        | 101                            | H'00000194 to<br>H'00000197        | 7            | _                | _                      | -                |
|                         | SINT6  |        | 102                            | H'00000198 to<br>H'0000019B        | 6            | _                | _                      | -                |
|                         | SINT5  |        | 103                            | H'0000019C to<br>H'0000019F        | 5            | _                | _                      | -                |
|                         | SINT4  |        | 104                            | H'000001A0 to<br>H'000001A3        | 4            | _                | _                      | -                |
|                         | SINT3  |        | 105                            | H'000001A4 to<br>H'000001A7        | 3            | _                | _                      | _                |
|                         | SINT2  |        | 106                            | H'000001A8 to<br>H'000001AB        | 2            | _                | _                      | -                |
|                         | SINT1  |        | 107                            | H'000001AC to<br>H'000001AF        | 1            | _                | _                      | -                |
| DMAC                    | DMAC0  | DEI0   | 108                            | H'000001B0 to<br>H'000001B3        | 0 to 15 (0)  | IPR03 (15 to 12) | 1                      | -                |
|                         |        | HEI0   | 109                            | H'000001B4 to<br>H'000001B7        | <del>-</del> |                  | 2                      | _                |
|                         | DMAC1  | DEI1   | 112                            | H'000001C0 to<br>H'000001C3        | 0 to 15 (0)  | IPR03 (11 to 8)  | 1                      | -                |
|                         |        | HEI1   | 113                            | H'000001C4 to<br>H'000001C7        | _            |                  | 2                      |                  |
|                         | DMAC2  | DEI2   | 116                            | H'000001D0 to<br>H'000001D3        | 0 to 15 (0)  | IPR03 (7 to 4)   | 1                      | _                |
|                         |        | HEI2   | 117                            | H'000001D4 to<br>H'000001D7        | =            |                  | 2                      | Low              |

|                         |       |        | Interrupt Vector               |                                    |                         |                               | Priority            |                  |
|-------------------------|-------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|------------------|
| Interrupt Source Number |       | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |                  |
| DMAC                    | DMAC3 | DEI3   | 120                            | H'000001E0 to<br>H'000001E3        | 0 to 15 (0)             | IPR03 (3 to 0)                | 1                   | High<br><b>♠</b> |
|                         |       | HEI3   | 121                            | H'000001E4 to<br>H'000001E7        | _                       |                               | 2                   |                  |
|                         | DMAC4 | DEI4   | 124                            | H'000001F0 to<br>H'000001F3        | 0 to 15 (0)             | IPR04 (15 to 12)              | 1                   |                  |
|                         |       | HEI4   | 125                            | H'000001F4 to<br>H'000001F7        | -                       |                               | 2                   |                  |
|                         | DMAC5 | DEI5   | 128                            | H'00000200 to<br>H'00000203        | 0 to 15 (0)             | IPR04 (11 to 8)               | 1                   | -                |
|                         |       | HEI5   | 129                            | H'00000204 to<br>H'00000207        | -                       |                               | 2                   |                  |
|                         | DMAC6 | DEI6   | 132                            | H'00000210 to<br>H'00000213        | 0 to 15 (0)             | IPR04 (7 to 4)                | 1                   |                  |
|                         |       | HEI6   | 133                            | H'00000214 to<br>H'00000217        | -                       |                               | 2                   |                  |
|                         | DMAC7 | DEI7   | 136                            | H'00000220 to<br>H'00000223        | 0 to 15 (0)             | IPR04 (3 to 0)                | 1                   |                  |
|                         |       | HEI7   | 137                            | H'00000224 to<br>H'00000227        | _                       |                               | 2                   |                  |
| CMT                     | CMI0  |        | 140                            | H'00000230 to<br>H'00000233        | 0 to 15 (0)             | IPR05 (15 to 12)              | _                   |                  |
|                         | CMI1  |        | 144                            | H'00000240 to<br>H'00000243        | 0 to 15 (0)             | IPR05 (11 to 8)               | _                   |                  |
| WDT                     | ITI   |        | 148                            | H'00000250 to<br>H'00000253        | 0 to 15 (0)             | IPR05 (3 to 0)                | _                   | -                |
| ATU-A                   | ICIA0 |        | 152                            | H'00000260 to<br>H'00000263        | 0 to 15 (0)             | IPR06 (15 to 12)              | 1                   |                  |
|                         | ICIA1 |        | 153                            | H'00000264 to<br>H'00000267        | _                       |                               | 2                   |                  |
|                         | ICIA2 |        | 156                            | H'00000270 to<br>H'00000273        | 0 to 15 (0)             | IPR06 (11 to 8)               | 1                   | -                |
|                         | ICIA3 |        | 157                            | H'00000274 to<br>H'00000277        | _                       |                               | 2                   | ₩<br>Low         |

|         |                         | Interrupt Vector |        | - 1 4 4                        |                                    | Priority                |                               |                     |
|---------|-------------------------|------------------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interru | Interrupt Source Number |                  | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ATU-A   | ICIA4                   |                  | 160    | H'00000280 to<br>H'00000283    | 0 to 15 (0)                        | IPR06 (7 to 4)          | 1                             | High                |
|         | ICIA5                   |                  | 161    | H'00000284 to<br>H'00000287    | -                                  |                         | 2                             |                     |
|         | OVIA                    |                  | 164    | H'00000290 to<br>H'00000293    | 0 to 15 (0)                        | IPR06 (3 to 0)          | _                             | -                   |
| ATU-B   | CMIB0                   |                  | 168    | H'000002A0 to<br>H'000002A3    | 0 to 15 (0)                        | IPR07 (15 to 12)        | 1                             |                     |
|         | CMIB1                   |                  | 169    | H'000002A4 to<br>H'000002A7    | -                                  |                         | 2                             | -                   |
|         | CMIB6                   |                  | 172    | H'000002B0 to<br>H'000002B3    | 0 to 15 (0)                        | IPR07 (11 to 8)         | 1                             | -                   |
|         | ICIB0                   |                  | 173    | H'000002B4 to<br>H'000002B7    | <del>-</del>                       |                         | 2                             | _                   |
| ATU-C   | ATU-C0                  | IMIC00           | 176    | H'000002C0 to<br>H'000002C3    | 0 to 15 (0)                        | IPR07 (7 to 4)          | 1                             | _                   |
|         |                         | IMIC01           | 177    | H'000002C4 to<br>H'000002C7    | _                                  |                         | 2                             |                     |
|         |                         | IMIC02           | 178    | H'000002C8 to<br>H'000002CB    | _                                  |                         | 3                             | -                   |
|         |                         | IMIC03           | 179    | H'000002CC to<br>H'000002CF    | -                                  |                         | 4                             |                     |
|         |                         | OVIC0            | 180    | H'000002D0 to<br>H'000002D3    | 0 to 15 (0)                        | IPR07 (3 to 0)          | _                             |                     |
|         | ATU-C1                  | IMIC10           | 184    | H'000002E0 to<br>H'000002E3    | 0 to 15 (0)                        | IPR08 (15 to 12)        | 1                             | -                   |
|         |                         | IMIC11           | 185    | H'000002E4 to<br>H'000002E7    | -                                  |                         | 2                             | -                   |
|         |                         | IMIC12           | 186    | H'000002E8 to<br>H'000002EB    | -                                  |                         | 3                             | -                   |
|         |                         | IMIC13           | 187    | H'000002EC to<br>H'000002EF    | -                                  |                         | 4                             |                     |
|         |                         | OVIC1            | 188    | H'000002F0 to<br>H'000002F3    | 0 to 15 (0)                        | IPR08 (11 to 8)         |                               | Low                 |

|                  |        | Interrupt Vector |                                | - 1 ( (                            |                         | Priority<br>within IPR |                     |
|------------------|--------|------------------|--------------------------------|------------------------------------|-------------------------|------------------------|---------------------|
| Interrupt Source | Number | Vector           | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | Setting<br>Unit        | Default<br>Priority |
| ATU-C ATU-C2     | IMIC20 | 192              | H'00000300 to<br>H'00000303    | 0 to 15 (0)                        | IPR08 (7 to 4)          | 1                      | High<br><b>↑</b>    |
|                  | IMIC21 | 193              | H'00000304 to<br>H'00000307    | -                                  |                         | 2                      | _                   |
|                  | IMIC22 | 194              | H'00000308 to<br>H'0000030B    | <del>-</del>                       |                         | 3                      | _                   |
|                  | IMIC23 | 195              | H'0000030C to<br>H'0000030F    | _                                  |                         | 4                      | -                   |
|                  | OVIC2  | 196              | H'00000310 to<br>H'00000313    | 0 to 15 (0)                        | IPR08 (3 to 0)          | _                      | -                   |
| ATU-C3           | IMIC30 | 200              | H'00000320 to<br>H'00000323    | 0 to 15 (0)                        | IPR09 (15 to 12)        | 1                      | -                   |
|                  | IMIC31 | 201              | H'00000324 to<br>H'00000327    | -                                  |                         | 2                      | -                   |
|                  | IMIC32 | 202              | H'00000328 to<br>H'0000032B    |                                    |                         | 3                      |                     |
|                  | IMIC33 | 203              | H'0000032C to<br>H'0000032F    | _                                  |                         | 4                      | -                   |
|                  | OVIC3  | 204              | H'00000330 to<br>H'00000333    | 0 to 15 (0)                        | IPR09 (11 to 8)         | _                      | -                   |
| ATU-C4           | IMIC40 | 208              | H'00000340 to<br>H'00000343    | 0 to 15 (0)                        | IPR09 (7 to 4)          | 1                      | -                   |
|                  | IMIC41 | 209              | H'00000344 to<br>H'00000347    | <del>-</del>                       |                         | 2                      |                     |
|                  | IMIC42 | 210              | H'00000348 to<br>H'0000034B    | <del>-</del>                       |                         | 3                      | _                   |
|                  | IMIC43 | 211              | H'0000034C to<br>H'0000034F    | _                                  |                         | 4                      |                     |
|                  | OVIC4  | 212              | H'00000350 to<br>H'00000353    | 0 to 15 (0)                        | IPR09 (3 to 0)          |                        | Low                 |

|                  | Interr |        |                                |                                    |                         | Priority                      |                     |
|------------------|--------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interrupt Source | Number | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ATU-D ATU-D0     | CMID00 | 216    | H'00000360 to<br>H'00000363    | 0 to 15 (0)                        | IPR10 (15 to 12)        | 1                             | High<br><b>∱</b>    |
|                  | CMID01 | 217    | H'00000364 to<br>H'00000367    | _                                  |                         | 2                             | -<br>-<br>-         |
|                  | CMID02 | 218    | H'00000368 to<br>H'0000036B    | _                                  |                         | 3                             |                     |
|                  | CMID03 | 219    | H'0000036C to<br>H'0000036F    | -                                  |                         | 4                             |                     |
|                  | OVI1D0 | 220    | H'00000370 to<br>H'00000373    | 0 to 15 (0)                        | IPR10 (11 to 8)         | 1                             |                     |
|                  | OVI2D0 | 221    | H'00000374 to<br>H'00000377    | _                                  |                         | 2                             |                     |
|                  | UDID00 | 224    | H'00000380 to<br>H'00000383    | 0 to 15 (0)                        | IPR10 (7 to 4)          | 1                             |                     |
|                  | UDID01 | 225    | H'00000384 to<br>H'00000387    | _                                  |                         | 2                             |                     |
|                  | UDID02 | 226    | H'00000388 to<br>H'0000038B    | _                                  |                         | 3                             | -<br>-<br>-         |
|                  | UDID03 | 227    | H'0000038C to<br>H'0000038F    | _                                  |                         | 4                             |                     |
| ATU-D1           | CMID10 | 228    | H'00000390 to<br>H'00000393    | 0 to 15 (0)                        | IPR10 (3 to 0)          | 1                             |                     |
|                  | CMID11 | 229    | H'00000394 to<br>H'00000397    | _                                  |                         | 2                             |                     |
|                  | CMID12 | 230    | H'00000398 to<br>H'0000039B    | _                                  |                         | 3                             |                     |
|                  | CMID13 | 231    | H'0000039C to<br>H'0000039F    |                                    |                         | 4                             |                     |
|                  | OVI1D1 | 232    | H'000003A0 to<br>H'000003A3    | 0 to 15 (0)                        | IPR11 (15 to 12)        | 1                             |                     |
|                  | OVI2D1 | 233    | H'000003A4 to<br>H'000003A7    |                                    |                         | 2                             | Low                 |

|                  |        | Interrupt Vector |                                |                                    |  | Priority                      |                     |
|------------------|--------|------------------|--------------------------------|------------------------------------|--|-------------------------------|---------------------|
| Interrupt Source | Number | Vector           | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit)                          | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ATU-D ATU-D1     | UDID10 | 236              | H'000003B0 to<br>H'000003B3    | 0 to 15 (0)                        | IPR11 (11 to 8)                                  | 1                             | High<br><b>↑</b>    |
|                  | UDID11 | 237              | H'000003B4 to<br>H'000003B7    | _                                  |  | 2                             | -<br>-<br>-         |
|                  | UDID12 | 238              | H'000003B8 to<br>H'000003BB    | _                                  |  | 3                             |                     |
|                  | UDID13 | 239              | H'000003BC to<br>H'000003BF    | -                                  |  | 4                             |                     |
| ATU-D2           | CMID20 | 240              | H'000003C0 to<br>H'000003C3    | 0 to 15 (0)                        | IPR11 (7 to 4)  IPR11 (3 to 0)  IPR12 (15 to 12) | 1                             |                     |
|                  | CMID21 | 241              | H'000003C4 to<br>H'000003C7    | 0 to 15 (0)                        |  | 2                             |                     |
|                  | CMID22 | 242              | H'000003C8 to<br>H'000003CB    |                                    |  | 3                             | -                   |
|                  | CMID23 | 243              | H'000003CC to<br>H'000003CF    |                                    |  | 4                             | -                   |
|                  | OVI1D2 | 244              | H'000003D0 to<br>H'000003D3    |                                    |  | 1                             | -                   |
|                  | OVI2D2 | 245              | H'000003D4 to<br>H'000003D7    | _                                  |  | 2                             |                     |
|                  | UDID20 | 248              | H'000003E0 to<br>H'000003E3    | 0 to 15 (0)                        |  | 1                             |                     |
|                  | UDID21 | 249              | H'000003E4 to<br>H'000003E7    | =                                  |  | 2                             |                     |
|                  | UDID22 | 250              | H'000003E8 to<br>H'000003EB    | _                                  |  | 3                             | _                   |
|                  | UDID23 | 251              | H'000003EC to<br>H'000003EF    | _                                  |  | 4                             | Low                 |

|                  |        | Interrupt Vector |                                |                                    |                         | Priority                      |                     |
|------------------|--------|------------------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interrupt Source | Number | Vector           | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ATU-D ATU-D3     | CMID30 | 252              | H'000003F0 to<br>H'000003F3    | 0 to 15 (0)                        | IPR12 (11 to 8)         | 1                             | High<br><b>∱</b>    |
|                  | CMID31 | 253              | H'000003F4 to<br>H'000003F7    | •                                  |                         | 2                             | -<br>-<br>-         |
|                  | CMID32 | 254              | H'000003F8 to<br>H'000003FB    | _                                  |                         | 3                             |                     |
|                  | CMID33 | 255              | H'000003FC to<br>H'000003FF    | _                                  |                         | 4                             |                     |
|                  | OVI1D3 | 256              | H'00000400 to<br>H'00000403    | 0 to 15 (0)                        | IPR12 (7 to 4)          | 1                             | _                   |
|                  | OVI2D3 | 257              | H'00000404 to<br>H'00000407    | _                                  |                         | 2                             |                     |
|                  | UDID30 | 260              | H'00000410 to<br>H'00000413    | 0 to 15 (0)                        | IPR12 (3 to 0)          | 1                             |                     |
|                  | UDID31 | 261              | H'00000414 to<br>H'00000417    | -                                  |                         | 2                             |                     |
|                  | UDID32 | 262              | H'00000418 to<br>H'0000041B    |                                    |                         | 3                             |                     |
|                  | UDID33 | 263              | H'0000041C to<br>H'0000041F    | _                                  |                         | 4                             |                     |
| ATU-E ATU-E0     | CMIE00 | 288              | H'00000480 to<br>H'00000483    | 0 to 15 (0)                        | IPR14 (7 to 4)          | 1                             |                     |
|                  | CMIE01 | 289              | H'00000484 to<br>H'00000487    | _                                  |                         | 2                             |                     |
|                  | CMIE02 | 290              | H'00000488 to<br>H'0000048B    | _                                  |                         | 3                             | _                   |
|                  | CMIE03 | 291              | H'0000048C to<br>H'0000048F    | _                                  |                         | 4                             | Low                 |

|          |           |        | Inte   | rrupt Vector                   | - 1 4 4                            |                         | Priority                      | ₹                   |
|----------|-----------|--------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interrup | ot Source | Number | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ATU-E    | ATU-E1    | CMIE10 | 292    | H'00000490 to<br>H'00000493    | 0 to 15 (0)                        | IPR14 (3 to 0)          | 1                             | High<br><b>∱</b>    |
|          |           | CMIE11 | 293    | H'00000494 to<br>H'00000497    | _                                  |                         | 2                             | -                   |
|          |           | CMIE12 | 294    | H'00000498 to<br>H'0000049B    | _                                  |                         | 3                             | -                   |
|          |           | CMIE13 | 295    | H'0000049C to<br>H'0000049F    | _                                  |                         | 4                             | -                   |
|          | ATU-E2    | CMIE20 | 296    | H'000004A0 to<br>H'000004A3    | 0 to 15 (0)                        | IPR15 (15 to 12)        | 1                             | -                   |
|          |           | CMIE21 | 297    | H'000004A4 to<br>H'000004A7    | _                                  |                         | 2                             | -                   |
|          |           | CMIE22 | 298    | H'000004A8 to<br>H'000004AB    | <del>-</del>                       |                         | 3                             | -                   |
|          |           | CMIE23 | 299    | H'000004AC to<br>H'000004AF    | <del>-</del>                       |                         | 4                             | -                   |
|          | ATU-E3    | CMIE30 | 300    | H'000004B0 to<br>H'000004B3    | 0 to 15 (0)                        | IPR15 (11 to 8)         | 1                             | -                   |
|          |           | CMIE31 | 301    | H'000004B4 to<br>H'000004B7    | <del>-</del>                       |                         | 2                             | -                   |
|          |           | CMIE32 | 302    | H'000004B8 to<br>H'000004BB    | <del>-</del>                       |                         | 3                             | -                   |
|          |           | CMIE33 | 303    | H'000004BC to<br>H'000004BF    | <del>-</del>                       |                         | 4                             | -                   |
|          | ATU-E4    | CMIE40 | 304    | H'000004C0 to<br>H'000004C3    | 0 to 15 (0)                        | IPR15 (7 to 4)          | 1                             | _                   |
|          |           | CMIE41 | 305    | H'000004C4 to<br>H'000004C7    | _                                  |                         | 2                             | _                   |
|          |           | CMIE42 | 306    | H'000004C8 to<br>H'000004CB    | _                                  |                         | 3                             | _                   |
|          |           | CMIE43 | 307    | H'000004CC to<br>H'000004CF    | -                                  |                         | 4                             | Low                 |

|         |           |        | Inte   | rrupt Vector                   |                                    |                         | Priority                      | R                   |
|---------|-----------|--------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interru | pt Source | Number | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ATU-E   | ATU-E5    | CMIE50 | 308    | H'000004D0 to<br>H'000004D3    | 0 to 15 (0)                        | IPR15 (3 to 0)          | 1                             | High                |
|         |           | CMIE51 | 309    | H'000004D4 to<br>H'000004D7    | _                                  |                         | 2                             | -                   |
|         |           | CMIE52 | 310    | H'000004D8 to<br>H'000004DB    | _                                  |                         | 3                             | -                   |
|         |           | CMIE53 | 311    | H'000004DC to<br>H'000004DF    | -                                  |                         | 4                             | -                   |
| ATU-F   | ICIF0     |        | 312    | H'000004E0 to<br>H'000004E3    | 0 to 15 (0)                        | IPR16 (15 to 12)        | 1                             | -                   |
|         | ICIF1     |        | 313    | H'000004E4 to<br>H'000004E7    | <del>-</del>                       |                         | 2                             | _                   |
|         | ICIF2     |        | 314    | H'000004E8 to<br>H'000004EB    | _                                  |                         | 3                             | -                   |
|         | ICIF3     |        | 315    | H'000004EC to<br>H'000004EF    | <del>-</del>                       |                         | 4                             | _                   |
|         | ICIF4     |        | 316    | H'000004F0 to<br>H'000004F3    | 0 to 15 (0)                        | IPR16 (11 to 8)         | 1                             | -                   |
|         | ICIF5     |        | 317    | H'000004F4 to<br>H'000004F7    | _                                  |                         | 2                             | -                   |
|         | ICIF6     |        | 318    | H'000004F8 to<br>H'000004FB    | _                                  |                         | 3                             | -                   |
|         | ICIF7     |        | 319    | H'000004FC to<br>H'000004FF    | _                                  |                         | 4                             | -                   |
|         | ICIF8     |        | 320    | H'00000500 to<br>H'00000503    | 0 to 15 (0)                        | IPR16 (7 to 4)          | 1                             | -                   |
|         | ICIF9     |        | 321    | H'00000504 to<br>H'00000507    | -                                  |                         | 2                             | _                   |
|         | ICIF10    |        | 322    | H'00000508 to<br>H'0000050B    | _                                  |                         | 3                             |                     |
|         | ICIF11    |        | 323    | H'0000050C to<br>H'0000050F    |                                    |                         | 4                             | ↓<br>Low            |

|         |                  | Inte   | rrupt Vector                   |                                    |                         | Priority                      |                     |
|---------|------------------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interru | pt Source Number | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ATU-F   | ICIF12           | 324    | H'00000510 to<br>H'00000513    | 0 to 15 (0)                        | IPR16 (3 to 0)          | 1                             | High<br><b>♠</b>    |
|         | ICIF13           | 325    | H'00000514 to<br>H'00000517    | -                                  |                         | 2                             | -                   |
|         | ICIF14           | 326    | H'00000518 to<br>H'0000051B    | -                                  |                         | 3                             | -                   |
|         | ICIF15           | 327    | H'0000051C to<br>H'0000051F    | -                                  |                         | 4                             | -                   |
|         | ICIF16           | 328    | H'00000520 to<br>H'00000523    | 0 to 15 (0)                        | IPR17 (15 to 12)        | 1                             | -                   |
|         | ICIF17           | 329    | H'00000524 to<br>H'00000527    | =                                  |                         | 2                             | -                   |
|         | ICIF18           | 330    | H'00000528 to<br>H'0000052B    | _                                  |                         | 3                             | -                   |
|         | ICIF19           | 331    | H'0000052C to<br>H'0000052F    | _                                  |                         | 4                             | -                   |
|         | OVIF0            | 340    | H'00000550 to<br>H'00000553    | 0 to 15 (0)                        | IPR18 (15 to 12)        | 1                             | -                   |
|         | OVIF1            | 341    | H'00000554 to<br>H'00000557    | _                                  |                         | 2                             | -                   |
|         | OVIF2            | 342    | H'00000558 to<br>H'0000055B    | _                                  |                         | 3                             | -                   |
|         | OVIF3            | 343    | H'0000055C to<br>H'0000055F    | <del>-</del>                       |                         | 4                             | -                   |
|         | OVIF4            | 344    | H'00000560 to<br>H'00000563    | 0 to 15 (0)                        | IPR18 (11 to 8)         | 1                             | -                   |
|         | OVIF5            | 345    | H'00000564 to<br>H'00000567    | _                                  |                         | 2                             | -                   |
|         | OVIF6            | 346    | H'00000568 to<br>H'0000056B    | _                                  |                         | 3                             |                     |
|         | OVIF7            | 347    | H'0000056C to<br>H'0000056F    | _                                  |                         | 4                             | Low                 |

|         | Interrupt Vector |        |                                | Priority                           | ₹                       |                               |                     |
|---------|------------------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interru | pt Source Number | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ATU-F   | OVIF8            | 348    | H'00000570 to<br>H'00000573    | 0 to 15 (0)                        | IPR18 (7 to 4)          | 1                             | High<br><b>∱</b>    |
|         | OVIF9            | 349    | H'00000574 to<br>H'00000577    | _                                  |                         | 2                             | _                   |
|         | OVIF10           | 350    | H'00000578 to<br>H'0000057B    | _                                  |                         | 3                             | _                   |
|         | OVIF11           | 351    | H'0000057C to<br>H'0000057F    | -                                  |                         | 4                             | _                   |
|         | OVIF12           | 352    | H'00000580 to<br>H'00000583    | 0 to 15 (0)                        | IPR18 (3 to 0)          | 1                             | _                   |
|         | OVIF13           | 353    | H'00000584 to<br>H'00000587    | _                                  |                         | 2                             | _                   |
|         | OVIF14           | 354    | H'00000588 to<br>H'0000058B    | _                                  |                         | 3                             | _                   |
|         | OVIF15           | 355    | H'0000058C to<br>H'0000058F    | _                                  |                         | 4                             | _                   |
|         | OVIF16           | 356    | H'00000590 to<br>H'00000593    | 0 to 15 (0)                        | IPR19 (15 to 12)        | 1                             | _                   |
|         | OVIF17           | 357    | H'00000594 to<br>H'00000597    | _                                  |                         | 2                             | _                   |
|         | OVIF18           | 358    | H'00000598 to<br>H'0000059B    | _                                  |                         | 3                             | _                   |
|         | OVIF19           | 359    | H'0000059C to<br>H'0000059F    | _                                  |                         | 4                             | _                   |
| ATU-G   | CMIG0            | 368    | H'000005C0 to<br>H'000005C3    | 0 to 15 (0)                        | IPR20 (15 to 12)        | 1                             | _                   |
|         | CMIG1            | 369    | H'000005C4 to<br>H'000005C7    | _                                  |                         | 2                             | _                   |
|         | CMIG2            | 370    | H'000005C8 to<br>H'000005CB    | _                                  |                         | 3                             | _                   |
|         | CMIG3            | 371    | H'000005CC to<br>H'000005CF    | _                                  |                         | 4                             | Low                 |

|         |                  | Inte   | rrupt Vector                   |                                    |                         | Priority                      | ₹                   |
|---------|------------------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interru | pt Source Number | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ATU-G   | CMIG4            | 372    | H'000005D0 to<br>H'000005D3    | 0 to 15 (0)                        | IPR20 (11 to 8)         | 1                             | High<br><b>∱</b>    |
|         | CMIG5            | 373    | H'000005D4 to<br>H'000005D7    |                                    |                         | 2                             |                     |
| ATU-H   | CMIH             | 376    | H'000005E0 to<br>H'000005E3    |                                    | IPR20 (7 to 4)          |                               | _                   |
| ATU-J   | DFIJ0            | 380    | H'000005F0 to<br>H'000005F3    | 0 to 15 (0)                        | IPR21 (15 to 12)        | 1                             | _                   |
|         | DFIJ1            | 381    | H'000005F4 to<br>H'000005F7    | _                                  |                         | 2                             | _                   |
|         | OVIJ0            | 384    | H'00000600 to<br>H'00000603    | 0 to 15 (0)                        | IPR21 (11 to 8)         | 1                             | _                   |
|         | OVIJ1            | 385    | H'00000604 to<br>H'00000607    | <del>-</del>                       |                         | 2                             | _                   |
|         | DOVIJ0           | 388    | H'00000610 to<br>H'00000613    | 0 to 15 (0)                        | IPR21 (7 to 4)          | 1                             | _                   |
|         | DOVIJ1           | 389    | H'00000614 to<br>H'00000617    | _                                  |                         | 2                             | _                   |
| ADC     | ADI0             | 392    | H'00000620 to<br>H'00000623    | 0 to 15 (0)                        | IPR22 (15 to 12)        | _                             | _                   |
|         | ADI1             | 396    | H'00000630 to<br>H'00000633    | 0 to 15 (0)                        | IPR22 (11 to 8)         | _                             | _                   |
|         | ADID0            | 400    | H'00000640 to<br>H'00000643    | 0 to 15 (0)                        | IPR22 (7 to 4)          | 1                             | _                   |
|         | ADID1            | 401    | H'00000644 to<br>H'00000647    | _                                  |                         | 2                             | _                   |
|         | ADID2            | 402    | H'00000648 to<br>H'0000064B    | =                                  |                         | 3                             |                     |
|         | ADID3            | 403    | H'0000064C to<br>H'0000064F    | _                                  |                         | 4                             | Low                 |

|         |                   | Inte   | errupt Vector                  |                                    |                         | Priority                      |                     |
|---------|-------------------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interru | upt Source Number | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| ADC     | ADID4             | 404    | H'00000650 to<br>H'00000653    | 0 to 15 (0)                        | IPR22 (3 to 0)          | 1                             | High<br><b>♠</b>    |
|         | ADID5             | 405    | H'00000654 to<br>H'00000657    | -                                  |                         | 2                             | -                   |
|         | ADID6             | 406    | H'00000658 to<br>H'0000065B    | -                                  |                         | 3                             | -                   |
|         | ADID7             | 407    | H'0000065C to<br>H'0000065F    | _                                  |                         | 4                             | -                   |
|         | ADID8             | 408    | H'00000660 to<br>H'0000663     | 0 to 15 (0)                        | IPR23 (15 to 12)        | 1                             | -                   |
|         | ADID9             | 409    | H'00000664 to<br>H'0000667     | -                                  |                         | 2                             | -                   |
|         | ADID10            | 410    | H'00000668 to<br>H'000066B     | _                                  |                         | 3                             | -                   |
|         | ADID11            | 411    | H'0000066C to<br>H'0000066F    | _                                  |                         | 4                             | -                   |
|         | ADID12            | 412    | H'00000670 to<br>H'00000673    | 0 to 15 (0)                        | IPR23 (11 to 8)         | 1                             | -                   |
|         | ADID13            | 413    | H'00000674 to<br>H'00000677    | _                                  |                         | 2                             | -                   |
|         | ADID14            | 414    | H'00000678 to<br>H'0000067B    | _                                  |                         | 3                             | -                   |
|         | ADID15            | 415    | H'0000067C to<br>H'0000067F    | -                                  |                         | 4                             | -                   |
|         | ADID40            | 416    | H'00000680 to<br>H'0000683     | 0 to 15 (0)                        | IPR23 (7 to 4)          | _                             | -                   |
|         | ADID41            | 417    | H'00000684 to<br>H'0000687     | 0 to 15 (0)                        | IPR23 (3 to 0)          | _                             | -                   |
|         | ADID42            | 418    | H'00000688 to<br>H'000068B     | 0 to 15 (0)                        | IPR24 (15 to 12)        | _                             | -                   |
|         | ADID43            | 419    | H'0000068C to<br>H'000068F     | 0 to 15 (0)                        | IPR24 (11 to 8)         | _                             | -                   |
|         | ADID44            | 420    | H'00000690 to<br>H'00000693    | 0 to 15 (0)                        | IPR24 (7 to 4)          | _                             | Low                 |

|         |                |          | Inte   | rrupt Vector                   |                                    |                         | Priority                      | R                    |
|---------|----------------|----------|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|----------------------|
| Interru | upt Sourc      | e Number | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority  |
| ADC     | ADID45         |          | 421    | H'00000694 to<br>H'00000697    | 0 to 15 (0)                        | IPR24 (3 to 0)          | _                             | High<br><del>↑</del> |
|         | ADID46         |          | 422    | H'00000698 to<br>H'0000069B    | 0 to 15 (0)                        | IPR25 (15 to 12)        | _                             | _                    |
|         | ADID47         |          | 423    | H'0000069C to<br>H'0000069F    | 0 to 15 (0)                        | IPR25 (11 to 8)         | _                             | _                    |
| SCI     | SCI SCI_A ERIA |          | 424    | H'000006A0 to<br>H'000006A3    | 0 to 15 (0)                        | IPR26 (15 to 12)        | 1                             | _                    |
|         |                | RXIA     | 425    | H'000006A4 to<br>H'000006A7    | -                                  |                         | 2                             | _                    |
|         |                | TXIA     | 426    | H'000006A8 to<br>H'000006AB    | -                                  |                         | 3                             | _                    |
|         |                | TEIA     | 427    | H'000006AC to<br>H'000006AF    | -                                  |                         | 4                             | _                    |
|         | SCI_B          | ERIB     | 428    | H'000006B0 to<br>H'000006B3    | 0 to 15 (0)                        | IPR26 (11 to 8)         | 1                             | -                    |
|         |                | RXIB     | 429    | H'000006B4 to<br>H'000006B7    | -                                  |                         | 2                             | _                    |
|         |                | TXIB     | 430    | H'000006B8 to<br>H'000006BB    | -                                  |                         | 3                             | _                    |
|         |                | TEIB     | 431    | H'000006BC to<br>H'000006BF    | _                                  |                         | 4                             | _                    |
|         | SCI_C          | ERIC     | 432    | H'000006C0 to<br>H'000006C3    | 0 to 15 (0)                        | IPR26 (7 to 4)          | 1                             | _                    |
|         |                | RXIC     | 433    | H'000006C4 to<br>H'000006C7    | -                                  |                         | 2                             | _                    |
|         |                | TXIC     | 434    | H'000006C8 to<br>H'000006CB    | _                                  |                         | 3                             | -                    |
|         |                | TEIC     | 435    | H'000006CC to<br>H'000006CF    | _                                  |                         | 4                             | Low                  |

|         |           |          | Interrupt Vector Interrupt |                                |             | Priority                |                               |                     |
|---------|-----------|----------|----------------------------|--------------------------------|-------------|-------------------------|-------------------------------|---------------------|
| Interru | ıpt Sourc | e Number | Vector                     | Vector Table<br>Address Offset | Priority    | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| SCI     | SCI_D     | ERID     | 436                        | H'000006D0 to<br>H'000006D3    | 0 to 15 (0) | IPR26 (3 to 0)          | 1                             | High<br><b>♠</b>    |
|         |           | RXID     | 437                        | H'000006D4 to<br>H'000006D7    | -           |                         | 2                             | _                   |
|         |           | TXID     | 438                        | H'000006D8 to<br>H'000006DB    | -           |                         | 3                             | _                   |
|         |           | TEID     | 439                        | H'000006DC to<br>H'000006DF    | -           |                         | 4                             | _                   |
|         | SCI_E     | ERIE     | 440                        | H'000006E0 to<br>H'000006E3    | 0 to 15 (0) | IPR27 (15 to 12)        | 1                             | -                   |
|         |           | RXIE     | 441                        | H'000006E4 to<br>H'000006E7    | -           |                         | 2                             | _                   |
|         |           | TXIE     | 442                        | H'000006E8 to<br>H'000006EB    | _           |                         | 3                             | _                   |
|         |           | TEIE     | 443                        | H'000006EC to<br>H'000006EF    | _           |                         | 4                             | _                   |
| RSPI    | RSPI_A    | SPEIA    | 444                        | H'000006F0 to<br>H'000006F3    | 0 to 15 (0) | IPR27 (11 to 8)         | 1                             | _                   |
|         |           | SPRIA    | 445                        | H'000006F4 to<br>H'000006F7    | _           |                         | 2                             | _                   |
|         |           | SPTIA    | 446                        | H'000006F8 to<br>H'000006FB    | -           |                         | 3                             | _                   |
|         | RSPI_B    | SPEIB    | 448                        | H'00000700 to<br>H'00000703    | 0 to 15 (0) | IPR27 (7 to 4)          | 1                             | _                   |
|         |           | SPRIB    | 449                        | H'00000704 to<br>H'00000707    | _           |                         | 2                             | _                   |
|         |           | SPTIB    | 450                        | H'00000708 to<br>H'0000070B    | _           |                         | 3                             | _                   |
|         | RSPI_C    | SPEIC    | 452                        | H'00000710 to<br>H'00000713    | 0 to 15 (0) | IPR27 (3 to 0)          | 1                             | -                   |
|         |           | SPRIC    | 453                        | H'00000714 to<br>H'00000717    | _           |                         | 2                             | -                   |
|         |           | SPTIC    | 454                        | H'00000718 to<br>H'0000071B    | _           |                         | 3                             | Low                 |

|              |                                      |  | Inte   | errupt Vector                  |                                    |                         | Priority                      |                     |
|--------------|--------------------------------------|--|--------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|
| Interru      | pt Source I                          | Number   | Vector | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |
| RCAN<br>-TL1 | RCAN_A                               | ERSA   | 456    | H'00000720 to<br>H'00000723    | 0 to 15 (0)                        | IPR28 (15 to 12)        | 1                             | High<br><b>♠</b>    |
|              |                                      | OVRA   | 457    | H'00000724 to<br>H'00000727    |                                    |                         | 2                             |                     |
|              |                                      | RMA0   | 458    | H'00000728 to<br>H'0000072B    | <del>-</del>                       |                         | 3                             | -                   |
|              | RMA1 4                               |  | 459    | H'0000072C to<br>H'0000072F    | <del>-</del>                       |                         | 4                             | -                   |
|              | SLEA                                 |  | 460    | H'00000730 to<br>H'00000733    | <del>-</del>                       |                         | 5                             | -                   |
|              | MBEA 461 H'00000734 to<br>H'00000737 |  | _      |                                | 6                                  |                         |                               |                     |
|              | RCAN_B                               | B ERSB 464 H'00000740 to 0 to 15 (0) IPR28 (11 to 8)<br>H'00000743 |        | IPR28 (11 to 8)                | 1                                  |                         |                               |                     |
|              | OVRB 465                             |  | 465    | H'00000744 to<br>H'00000747    | _                                  |                         | 2                             |                     |
|              |                                      | RMB0   | 466    | H'00000748 to<br>H'0000074B    | _                                  |                         | 3                             |                     |
|              |                                      | RMB1   | 467    | H'0000074C to<br>H'0000074F    | _                                  |                         | 4                             |                     |
|              |                                      | SLEB   | 468    | H'00000750 to<br>H'00000753    | <del>-</del>                       |                         | 5                             | -                   |
|              |                                      | MBEB   | 469    | H'00000754 to<br>H'00000757    | _                                  |                         | 6                             |                     |
|              | RCAN_C                               | ERSC   | 472    | H'00000760 to<br>H'00000763    | 0 to 15 (0)                        | IPR28 (7 to 4)          | 1                             |                     |
|              |                                      | OVRC   | 473    | H'00000764 to<br>H'00000767    | _                                  |                         | 2                             |                     |
|              |                                      | RMC0   | 474    | H'00000768 to<br>H'0000076B    | -                                  |                         | 3                             | -                   |
|              |                                      | RMC1   | 475    | H'0000076C to<br>H'0000076F    | =                                  |                         | 4                             | -                   |
|              |                                      | SLEC   | 476    | H'00000770 to<br>H'00000773    | =                                  |                         | 5                             | -                   |
|              |                                      | MBEC   | 477    | H'00000774 to<br>H'00000777    | -                                  |                         | 6                             | Low                 |

|            |                         | Inte | rrupt Vector                   |                                    |                         | Priority                      |                     |  |
|------------|-------------------------|------|--------------------------------|------------------------------------|-------------------------|-------------------------------|---------------------|--|
| Interr     | Interrupt Source Number |      | Vector Table<br>Address Offset | Interrupt Priority (Initial Value) | Corresponding IPR (Bit) | within IPR<br>Setting<br>Unit | Default<br>Priority |  |
| A-<br>DMAC | TE74                    | 488  | H'000007A0 to<br>H'000007A3    | 0 to 15 (0)                        | IPR29 (15 to 12)        | 1                             | High<br>Low         |  |

## 8.6 Operation

## 8.6.1 Interrupt Operation Sequence

The sequence of interrupt operations is described below. Figure 8.2 shows the operation flow.

- 1. The interrupt request sources requests an interrupt to the interrupt controller.
- 2. The interrupt controller selects the highest-priority interrupt from the received interrupts according to the priority levels set in interrupt priority registers 01 to 29 (IPR01 to IPR29). Remaining interrupts are ignored\*. If multiple interrupts have the same IPR priority level or if multiple interrupts occur within a single module, the interrupt with the highest priority is selected according to the priority within the IPR setting unit and default priority shown in table 8.4.
- 3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt level mask bits (I3 to I0) in the status register (SR) of the CPU. If the interrupt request priority level is equal to or less than the level set in bits I3 to I0, the interrupt request is ignored. If the interrupt request priority level is higher than the level in bits I3 to I0, the interrupt controller receives the interrupt and requests an interrupt to the CPU.
- 4. The CPU detects the interrupt requested by the interrupt controller when the CPU decodes the instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception handling (figure 8.4).
- 5. The start address of the interrupt exception handler corresponding to the received interrupt is fetched from the exception handling vector table.
- 6. The contents of the status register (SR) are saved onto the stack, and the priority level of the received interrupt is copied to bits I3 to I0 in SR.
- 7. The contents of the program counter (PC) are saved onto the stack.
- 8. The CPU branches to the fetched start address of the interrupt exception handler and starts executing the program. The branch is not delayed branch.

Notes: The interrupt source flag should be cleared in the interrupt handler. The time from when the interrupt source flag is cleared to when the interrupt signal is negated in the CPU is the same time which is described as the time from occurrence of interrupt request until interrupt controller identifies priority in table 8.5. Therefore, read the interrupt source flag after clearing it so that an interrupt request that should have been cleared is not received again erroneously. After that, execute the RTE instruction.

\* Interrupt requests that are designated as edge-sensing are held pending until the interrupt requests are received. IRQ interrupts, however, can be cancelled by accessing the IRQ interrupt request register (IRQRR). For details, see section 8.4.3, IRQ Interrupts.

Interrupts held pending due to edge-sensing are cleared by a power-on reset.

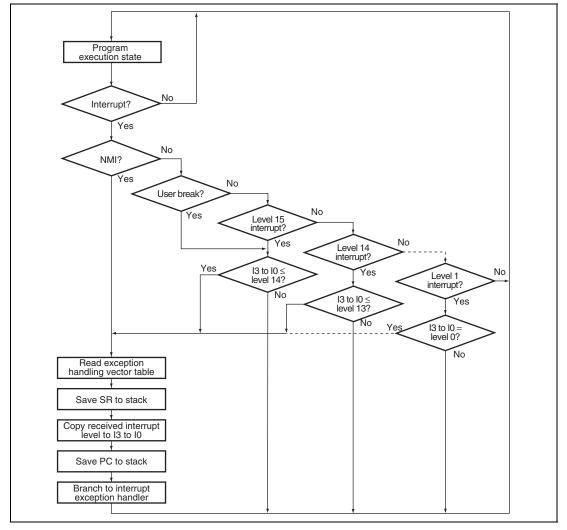


Figure 8.2 Interrupt Operation Flow

#### 8.6.2 **Stack after Interrupt Exception Handling**

Figure 8.3 shows the stack after interrupt exception handling.

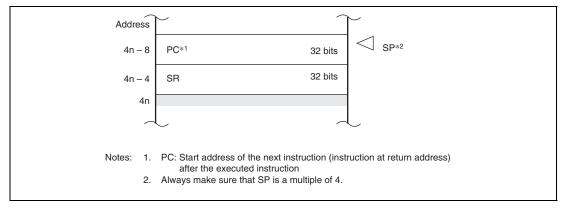


Figure 8.3 Stack after Interrupt Exception Handling

## 8.7 Interrupt Response Time

Table 8.5 lists the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception handling starts and fetching of the first instruction in the exception handler begins. The interrupt processing operations differ in the cases when banking is disabled, when banking is enabled without register bank overflow, and when banking is enabled with register bank overflow. Figures 8.4 and 8.5 show examples of pipeline operation when banking is disabled. Figures 8.6 and 8.7 show examples of pipeline operation when banking is enabled without register bank overflow. Figures 8.8 and 8.9 show examples of pipeline operation when banking is enabled with register bank overflow.

Note that in table 8.5, and figures 8.4 to 8.9, Icyc represents  $\phi$  cycle, and Bcyc and Pcyc represent P $\phi$  cycle.

**Table 8.5** Interrupt Response Time

|  |   |           |                    | N          | umber of Cy                     | cles               |                                |  |
|--|---|-----------|--------------------|------------|---------------------------------|--------------------|--------------------------------|--|
| Item   |   |           | NMI                | User Break | IRQ                             | SINT               | Peripheral<br>Module           | Remarks  |
| Time from occurre<br>until interrupt cont<br>compares it with n<br>interrupt request s | roller identifies p<br>nask bits in SR,         | oriority, | 2 lcyc +<br>3 Pcyc | 3 lcyc     | 2 lcyc +<br>1 Bcyc +<br>3 Pcyc  | 2 lcyc +<br>1 Bcyc | 2 lcyc +<br>1 Bcyc +<br>1 Pcyc |  |
| Time from input  | No register                                     | Min.      |                    | 3          | Min. is when the interrupt wait |                    |                                |  |
| of interrupt request signal to CPU until sequence currently being                      | banking   | Max.      |                    | 4 leye     | c + 2(m1 + m                    | n2) + m3           |                                | Time is zero. Max. is when a higher-priority interrupt request has occurred during interrupt exception handling. |
| executed is completed,   | Register  | Min.      | _                  | _          |                                 | 3 lcyc + m1 + m2   |                                | Min. is when the interrupt wait  |
| interrupt exception handling starts, and first instruction in                          | banking<br>without<br>register bank<br>overflow | Max.      | _                  | _          |                                 | 12 lcyc + m1       | + m2                           | Time is zero.  Max. is when an interrupt request has occurred during execution of the RESBANK instruction.       |
| interrupt  | Register  | Min.      | _                  | _          |                                 | 3 lcyc + m1        | + m2                           | Min. is when the interrupt wait  |
| exception handler<br>is fetched  | banking with<br>register bank<br>overflow       | Max.      | _                  | _          | 3 lcy                           | rc + m1 + m2       | + 19(m4)                       | Time is zero.  Max. is when an interrupt request has occurred during execution of the RESBANK instruction.       |

#### Number of States

| Item                    |  |      | NMI   | User Break                      | IRQ  | SINT   | Peripheral<br>Module                                    | Remarks   |
|-------------------------|--|------|---|---------------------------------|--|--|---|---|
| Interrupt response time | No register<br>banking                                   | Min. | 5 lcyc +<br>3 Pcyc +<br>m1 + m2             | 6 lcyc +<br>m1 + m2             | 5 lcyc +<br>1 Bcyc +<br>3 Pcyc +<br>m1 + m2              | 5 lcyc +<br>1 Bcyc +<br>m1 + m2              | 5 lcyc +<br>1 Bcyc +<br>1 Pcyc +<br>m1 + m2             | 200-MHz operation* $^{1}$ * $^{2}$ : 0.040 to 0.135 $\mu s$           |
|                         |  | Max. | 6 lcyc +<br>3 Pcyc +<br>2 (m1 + m2)<br>+ m3 | 7 lcyc +<br>2 (m1 + m2)<br>+ m3 | 6 lcyc +<br>1 Bcyc +<br>3 Pcyc +<br>2 (m1 + m2)<br>+ m3  | 6 lcyc +<br>1 Bcyc +<br>2 (m1 + m2)<br>+ m3  | 6 lcyc +<br>1 Bcyc +<br>1 Pcyc +<br>2 (m1 + m2)<br>+ m3 | 200-MHz operation* <sup>1</sup> * <sup>2</sup> :<br>0.060 to 0.155 μs |
|                         | Register<br>banking<br>without register<br>bank overflow | Min. | _   | _                               | 5 lcyc +<br>1 Bcyc +<br>3 Pcyc +<br>m1 + m2              | 5 lcyc +<br>1 Bcyc +<br>m1 + m2              | 5 lcyc +<br>1 Bcyc +<br>1 Pcyc +<br>m1 + m2             | 200-MHz operation* <sup>1</sup> * <sup>2</sup> : 0.060 to 0.135 μs    |
|                         |  | Max. | _   | _                               | 14 lcyc +<br>1 Bcyc +<br>3 Pcyc +<br>m1 + m2             | 14 lcyc +<br>1 Bcyc +<br>m1 + m2             | 14 lcyc +<br>1 Bcyc +<br>1 Pcyc +<br>m1 + m2            | 200-MHz operation* <sup>1</sup> * <sup>2</sup> :<br>0.105 to 0.180 μs |
|                         | Register<br>banking with<br>register bank<br>overflow    | Min. | _   | _                               | 5 lcyc +<br>1 Bcyc +<br>3 Pcyc +<br>m1 + m2              | 5 lcyc +<br>1 Bcyc +<br>m1 + m2              | 5 lcyc +<br>1 Bcyc +<br>1 Pcyc +<br>m1 + m2             | 200-MHz operation* <sup>1</sup> * <sup>2</sup> : 0.060 to 0.135 μs    |
|                         |  | Max. | —   | _                               | 5 lcyc +<br>1 Bcyc +<br>3 Pcyc +<br>m1 + m2 +<br>19 (m4) | 5 lcyc +<br>1 Bcyc +<br>m1 + m2 +<br>19 (m4) | 5 lcyc +<br>1 Bcyc +<br>1 Pcyc +<br>m1 +m2 +<br>19 (m4) | 200-MHz operation* <sup>1</sup> * <sup>2</sup> :<br>0.155 to 0.230 μs |

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

m1: Vector address read (longword read)

m2: SR saving (longword write)m3: PC saving (longword write)

m4: Restoring banked registers (R0 to R14, GBR, MACH, MACL, and PR) from the stack.

- 1. When m1 = m2 = m3 = m4 = 1 lcyc
- 2. When  $\phi$  and P $\phi$  are 200 and 40 MHz, respectively

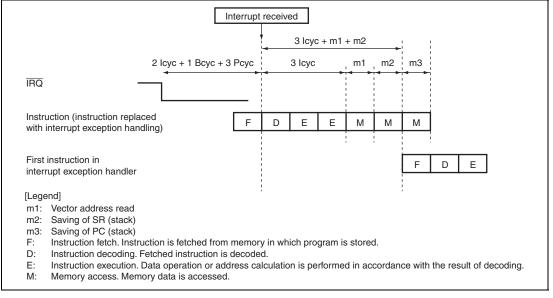


Figure 8.4 Example of Pipeline Operation when IRQ Interrupt is Received (No Register Banking)

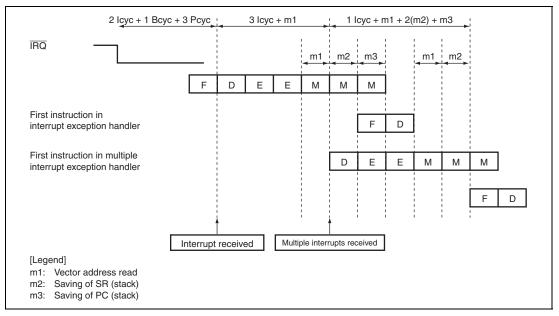


Figure 8.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

Page 165 of 1824

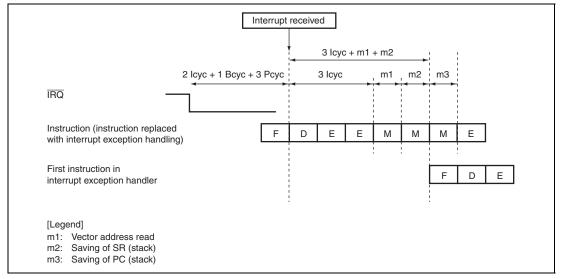


Figure 8.6 Example of Pipeline Operation when IRQ Interrupt is Received (Register Banking without Register Bank Overflow)

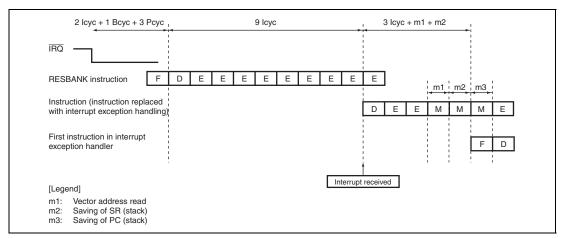


Figure 8.7 Example of Pipeline Operation when Interrupt is Received during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

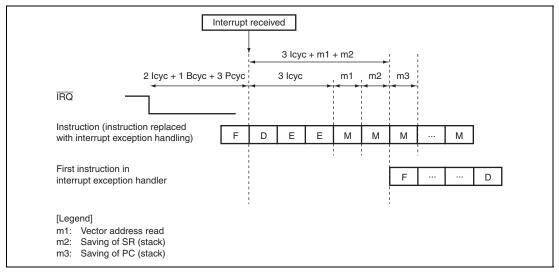


Figure 8.8 Example of Pipeline Operation when IRQ Interrupt is Received (Register Banking with Register Bank Overflow)

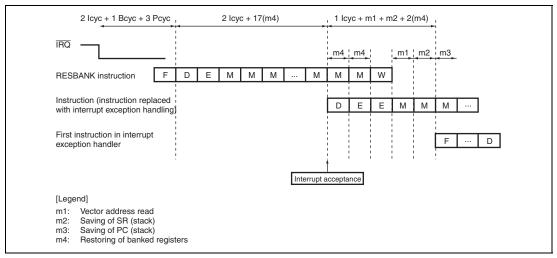


Figure 8.9 Example of Pipeline Operation when Interrupt is Received during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

Page 167 of 1824

## 8.8 Register Banks

This LSI has fifteen register banks used to save and restore registers for the interrupt processing at high speed. Figure 8.10 is the register bank configuration.

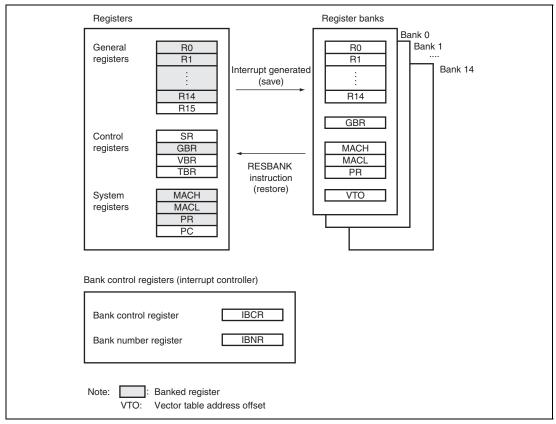


Figure 8.10 Overview of Register Bank Configuration

#### 8.8.1 Banked Registers and Input/Output Method

#### (1) Banked Registers

The contents of the general registers (R0 to R14), global base register (GBR), multiply and accumulate registers (MACH and MACL), and procedure register (PR), and the vector table address offset are banked.

#### (2) Input/Output Configuration

This LSI has fifteen register banks, bank 0 to bank 14. Register banks are stacked in first-in last-out (FILO) sequence. Saving takes place in the order of the bank number from 0 to 14 and restoring takes place in the reverse order from the last bank saved.

## 8.8.2 Bank Saving and Restoring Operations

#### (1) Saving to Bank

Figure 8.11 shows register bank saving operations. The following operations are performed when an interrupt for which usage of register banks is allowed is received by the CPU:

- (a) Assume that the bank number bit (BN) in the bank number register (IBNR) before the interrupt is generated is i.
- (b) The contents of registers R0 to R14, GBR, MACH, MACL, and PR, and the vector table address offset (VTO) of the received interrupt are saved in bank i.
- (c) The BN bit is incremented by 1.

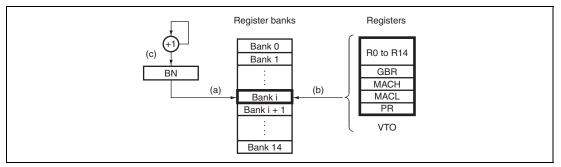


Figure 8.11 Bank Saving Operation

Figure 8.12 shows the timing for saving registers to a register bank. Saving registers to a register bank takes place between the start of interrupt exception handling and the start of fetching the first instruction in the interrupt exception handler.

In figure 8.12, Icyc represents the cycle of the  $\phi$  clock and Bcyc and Pcyc represent the cycle of the Po clock.

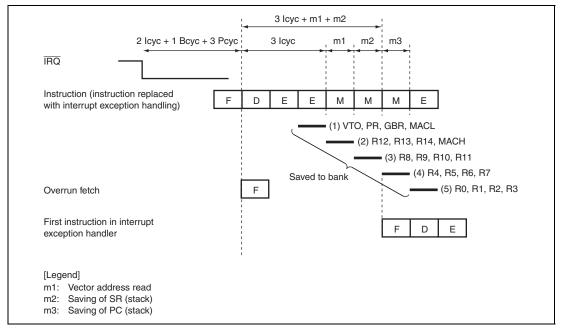


Figure 8.12 Bank Save Timing

#### **(2) Restoring from Bank**

The RESBANK (restoring from register bank) instruction is used to restore data saved in a register bank. After restoring data from the register banks with the RESBANK instruction at the end of the interrupt exception handler, execute the RTE instruction to return from the interrupt exception handler.

### 8.8.3 Saving and Restoring Operations after Saving Registers to All Banks

Assume that all register banks has been used for saving registers when an interrupt for which usage of the register banks is enabled is received by the CPU. When the BOVE bit in the bank number register (IBNR) is cleared to 0, registers are automatically saved to the stack area instead of saving to a register bank. When the BOVE bit in IBNR is set to 1, a register bank overflow exception occurs and registers are not saved to the stack area.

Saving and restoring operations when using the stack are shown below.

## (1) Saving to Stack

- 1. The status register (SR) and program counter (PC) are saved to the stack during interrupt exception handling.
- 2. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are saved to the stack. The registers are saved to the stack in the order of MACL, MACH, GBR, PR, R14, R13, ..., R1, and R0.
- 3. The register bank overflow bit (BO) in SR is set to 1.
- 4. The bank number bit (BN) value in the bank number register (IBNR) remains set to a maximum value of 15.

## (2) Restoring from Stack

Operations when the RESBANK (restoring from register bank) instruction is executed with the register bank overflow bit (BO) in SR set to 1 are shown below.

- 1. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack. The registers are restored from the stack in the order of R0, R1, ..., R13, R14, PR, GBR, MACH, and MACL.
- 2. The bank number bit (BN) value in the bank number register (IBNR) remains set to a maximum value of 15.

## 8.8.4 Register Bank Exception

There are two register bank exceptions (register bank errors): a register bank overflow and a register bank underflow.

#### (1) Register Bank Overflow

Assume that all register banks has been used for saving registers when an interrupt for which usage of the register banks is enabled is received by the CPU. When the BOVE bit in IBNR is set to 1, a register bank overflow exception occurs and registers are not saved to the stack area.

#### (2) Register Bank Underflow

This exception occurs if the RESBANK (restoring from register bank) instruction is executed when no data has been saved to the register banks. In this case, the contents of R0 to R14, GBR, MACH, MACL, and PR do not change. In addition, the bank number bit (BN) value in the bank number register (IBNR) remains set to 0.

### 8.8.5 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. When this happens, the CPU operation is shown below.

- 1. The start address of the exception handler for the register bank error is fetched from the exception handling vector table.
- 2. The status register (SR) is saved to the stack.
- 3. The program counter (PC) is saved to the stack. At this time, the start address of the instruction to be executed after the last executed instruction is in PC and is saved when a register bank overflow occurs. The start address of the executed RESBANK instruction is in PC and is saved when a register bank underflow occurs. To prevent multiple interrupts from occurring at a register bank overflow, the interrupt priority level that caused the register bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
- 4. Program execution starts from the exception handler start address.

## 8.9 Data Transfer with Interrupt Request Signals

Interrupt request signals can be used to activate the DMAC or A-DMAC for data transfer.

Interrupt sources that are set to activate the DMAC or A-DMAC are masked without being received in the INTC. The respective mask conditions for the DMAC and A-DMAC are shown below.

#### (1) Mask Condition for DMAC

```
Mask condition = DME • (DE0 • interrupt source select 0 + DE1 • interrupt source select 1 + DE2 • interrupt source select 2 + DE3 • interrupt source select 3 + DE4 • interrupt source select 4 + DE5 • interrupt source select 5 + DE6 • interrupt source select 6 + DE7 • interrupt source select 7)
```

Where DME is bit 0 in DMAOR of the DMAC and DEn (n = 0 to 7) is bit n in CHCR0 to CHCR7 of the DMAC.

## (2) Mask Condition for A-DMAC

Mask condition = NMI + (DME • DEn) ; For ADC or ATU-III

Mask condition = NMI + (DME • (TCRn + TEn)) ; For RSPI or SCI

Where NMI is bit 1 in DMAOR of the DMAC, DME is bit 0 in ADMAOR of the A-DMAC, DEn is a bit in ADMADE corresponding to channel n of the A-DMAC, TCRn is a bit in ADMATCR corresponding to channel n of the A-DMAC, and TEn is a bit in ADMATE corresponding to channel n of the A-DMAC. TCRn should not be 0.

Note that RCAN-TL1 transfer function by the A-DMAC does not use an interrupt request signal from the RCAN-TL1. Accordingly, even if it is specified to activate the A-DMAC, it is not masked to be received in the INTC.

The DMAC or the A-DMAC clears the interrupt source flag of the interrupt request source after data transfer in response to the interrupt request signal.

For details, see section 11, Direct Memory Access Controller (DMAC), and section 12, Automotive Direct Memory Access Controller (A-DMAC).

Figure 8.13 is a block diagram of interrupt control.

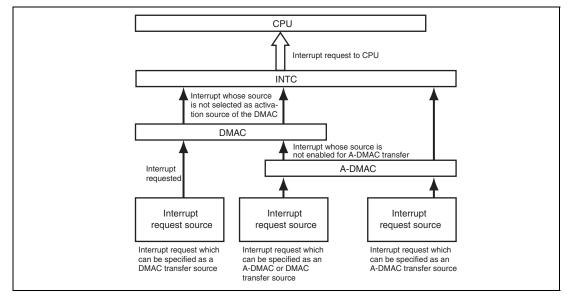


Figure 8.13 Interrupt Control Block Diagram

#### 8.9.1 Interrupt Request Signals as Sources for CPU (Not for Activating DMAC)

- 1. Do not select DMAC activating sources or clear the DME bit to 0. If DMAC activating sources are selected, clear the DE bit to 0 for the relevant channel of the DMAC.
- 2. When an interrupt occurs, the interrupt is requested to the CPU.
- 3. Clear the interrupt source in the interrupt exception handler. The CPU executes the necessary processing.

#### 8.9.2 Interrupt Request Signals as Sources for Activating DMAC (Not for CPU)

- 1. Selects the activation source in the DMAC and set both the DME and DE bits to 1. This settings mask CPU interrupt sources masked regardless of the interrupt priority register settings.
- 2. An activating signal is sent to the DMAC when an interrupt occurs.
- 3. The DMAC clears the interrupt source when starting transfer.

### 8.9.3 Interrupt Request Signals as Sources for Activating A-DMAC (Not for CPU)

- 1. Set the DME bit in the A-DMAC to 1, and the DE bit in the appropriate channel to 1 or ADMATCR to other than 0 (the number of transfers). This settings mask CPU interrupt sources masked regardless of the interrupt priority register settings.
- 2. An activating signal is sent to the A-DMAC when an interrupt occurs.
- 3. The A-DMAC clears the interrupt sources when starting transfer.

## 8.10 Usage Note

Page 174 of 1824

### 8.10.1 Timing to Clear Interrupt Source

Clear the interrupt source flag to 0 in the interrupt exception handler. The time described as "Time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU" in table 8.5 is required until the interrupt is cleared in the CPU after clearing the interrupt source flag to 0. Perform dummy read the interrupt source flag after clearing it to ensure that the interrupt request that should have been cleared is not received again erroneously. After that, the RTE instruction can be executed and the interrupt is not received again erroneously. To change interrupt levels using the LDC instruction, execute NOP at least three times after the dummy read of the interrupt source flag, and then execute the LDC instruction.

# Section 9 User Break Controller (UBC)

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design an effective self-monitoring debugger, enabling the chip to debug programs without using an in-circuit emulator. Instruction fetch or data read/write (bus master (CPU, DMAC, or A-DMAC) selection in the case of data read/write), data size, address value, and stop timing in the case of instruction fetch are break conditions that can be set in the UBC. Since this LSI uses a Harvard architecture, instruction fetch on the CPU bus (C bus) is performed by issuing bus cycles on the instruction fetch bus (F bus), and data access on the C bus is performed by issuing bus cycles on the memory access bus (M bus). The UBC monitors the C bus and internal bus (I bus).

## 9.1 Features

1. The following break comparison conditions can be set.

Number of break channels: four channels (channels 0 to 3)

User break can be requested as the independent condition on channels 0 to 3, respectively.

Address

Comparison of the 32-bit address is maskable in 1-bit units.

One of the three address buses (F address bus (FAB), M address bus (MAB), and I address bus (IAB)) can be selected.

- Bus master when I bus is selected
  - Selection of CPU cycles, DMAC cycles, or A-DMAC cycles
- Bus cycle
  - Instruction fetch (only when C bus is selected) or data access
- Read/write
- Operand size

Byte, word, and longword

- 2. A user-designed user-break condition exception handling routine can be run.
- 3. In an instruction fetch cycle, it can be selected whether a break is set before or after an instruction is executed.
- 4. When a break condition is satisfied, a trigger signal is output from the UBCTRG pin.

Figure 9.1 shows a block diagram of the UBC.

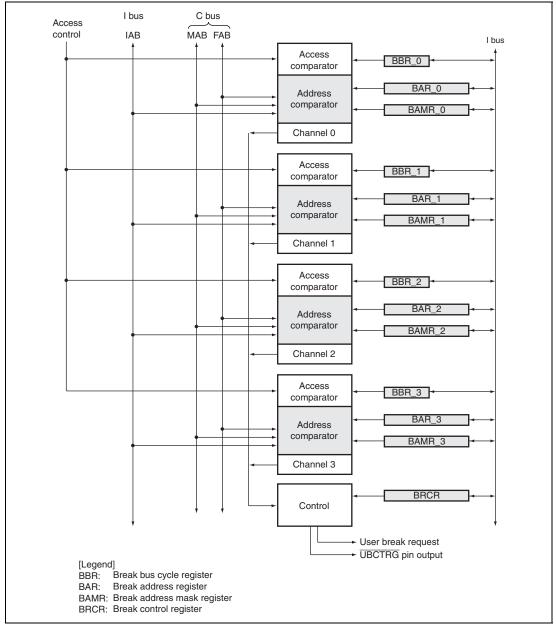


Figure 9.1 Block Diagram of UBC

# 9.2 Input/Output Pin

Table 9.1 shows the pin configuration of the UBC.

**Table 9.1** Pin Configuration

| Pin Name    | Symbol | I/O    | Function   |
|-------------|--------|--------|--|
| UBC trigger | UBCTRG | Output | Indicates that a setting condition is satisfied on either of channels 0 to 3 of the UBC. |

# 9.3 Register Descriptions

The UBC has the following registers.

**Table 9.2** Register Configuration

| Channel | Register Name                 | Symbol | R/W | Initial Value | Address    | Access<br>Size |
|---------|-------------------------------|--------|-----|---------------|------------|----------------|
| 0       | Break address register_0      | BAR_0  | R/W | H'00000000    | H'FFFC0400 | 32             |
|         | Break address mask register_0 | BAMR_0 | R/W | H'00000000    | H'FFFC0404 | 32             |
|         | Break bus cycle register_0    | BBR_0  | R/W | H'0000        | H'FFFC04A0 | 16             |
| 1       | Break address register_1      | BAR_1  | R/W | H'00000000    | H'FFFC0410 | 32             |
|         | Break address mask register_1 | BAMR_1 | R/W | H'00000000    | H'FFFC0414 | 32             |
|         | Break bus cycle register_1    | BBR_1  | R/W | H'0000        | H'FFFC04B0 | 16             |
| 2       | Break address register_2      | BAR_2  | R/W | H'00000000    | H'FFFC0420 | 32             |
|         | Break address mask register_2 | BAMR_2 | R/W | H'00000000    | H'FFFC0424 | 32             |
|         | Break bus cycle register_2    | BBR_2  | R/W | H'0000        | H'FFFC04A4 | 16             |
| 3       | Break address register_3      | BAR_3  | R/W | H'00000000    | H'FFFC0430 | 32             |
|         | Break address mask register_3 | BAMR_3 | R/W | H'00000000    | H'FFFC0434 | 32             |
|         | Break bus cycle register_3    | BBR_3  | R/W | H'0000        | H'FFFC04B4 | 16             |
| Common  | Break control register        | BRCR   | R/W | H'00000000    | H'FFFC04C0 | 8, 32          |

## 9.3.1 Break Address Register\_0 (BAR\_0)

BAR\_0 is a 32-bit readable/writable register. BAR\_0 specifies the address used as a break condition in channel 0. BAR\_0 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 31         | 30         | 29         | 28         | 27         | 26         | 25         | 24         | 23         | 22         | 21         | 20         | 19         | 18         | 17         | 16         |
|----------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | BA0_<br>31 | BA0_<br>30 | BA0_<br>29 | BA0_<br>28 | BA0_<br>27 | BA0_<br>26 | BA0_<br>25 | BA0_<br>24 | BA0_<br>23 | BA0_<br>22 | BA0_<br>21 | BA0_<br>20 | BA0_<br>19 | BA0_<br>18 | BA0_<br>17 | BA0_<br>16 |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W        |
|                |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |
| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|                | BA0_<br>15 | BA0_<br>14 | BA0_<br>13 | BA0_<br>12 | BA0_<br>11 | BA0_<br>10 | BA0_<br>9  | BA0_<br>8  | BA0_<br>7  | BA0_<br>6  | BA0_<br>5  | BA0_<br>4  | BA0_<br>3  | BA0_<br>2  | BA0_<br>1  | BA0_<br>0  |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W        |

| Bit     | Bit Name  | Initial<br>Value | R/W | Description   |
|---------|-----------|------------------|-----|---|
| 31 to 0 | BA0_31 to | All 0            | R/W | Break Address 0   |
|         | BA0_0     |                  |     | Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 0.             |
|         |           |                  |     | When the C bus and instruction fetch cycle are selected by BBR_0, specify an FAB address in bits BA0_31 to BA0_0. |
|         |           |                  |     | When the C bus and data access cycle are selected by BBR_0, specify an MAB address in bits BA0_31 to BA0_0.       |

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR\_0 to 0.

## 9.3.2 Break Address Mask Register\_0 (BAMR\_0)

BAMR\_0 is a 32-bit readable/writable register. BAMR\_0 specifies bits masked in the break address bits specified by BAR\_0. BAMR\_0 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 31          | 30          | 29          | 28          | 27          | 26          | 25          | 24          | 23          | 22          | 21          | 20          | 19          | 18          | 17          | 16          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | BAM0_<br>31 | BAM0_<br>30 | BAM0_<br>29 | BAM0_<br>28 | BAM0_<br>27 | BAM0_<br>26 | BAM0_<br>25 | BAM0_<br>24 | BAM0_<br>23 | BAM0_<br>22 | BAM0_<br>21 | BAM0_<br>20 | BAM0_<br>19 | BAM0_<br>18 | BAM0_<br>17 | BAM0_<br>16 |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |
|                |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9           | 8           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|                | BAM0_       |             |             |             | BAM0_       | BAM0_       |             |             | BAM0_       |             |             | 1 4 -       | BAM0_       | BAM0_       | BAM0_       | BAM0_       |
|                | 15          | 14          | 13          | 12          | 11          | 10          | 9           | 8           | /           | 6           | 5           | 4           | 3           | 2           | ı           | 0           |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |

|         |            | Initial |     |   |
|---------|------------|---------|-----|---|
| Bit     | Bit Name   | Value   | R/W | Description   |
| 31 to 0 | BAM0_31 to | All 0   | R/W | Break Address Mask 0  |
|         | BAM0_0     |         |     | Specify bits masked in the channel 0 break address bits specified by BAR_0 (BA0_31 to BA0_0).     |
|         |            |         |     | Break address bit BA0_n is included in the break condition  |
|         |            |         |     | <ol> <li>Break address bit BA0_n is masked and not<br/>included in the break condition</li> </ol> |
|         |            |         |     | Note: $n = 31 \text{ to } 0$  |

## 9.3.3 Break Bus Cycle Register\_0 (BBR\_0)

BBR\_0 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 0. BBR\_0 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 15 | 14 | 13    | 12 | 11 | 10         | 9   | 8   | 7              | 6   | 5   | 4     | 3    | 2     | 1    | 0     |
|----------------|----|----|-------|----|----|------------|-----|-----|----------------|-----|-----|-------|------|-------|------|-------|
|                | -  | -  | UBID0 | -  | -  | CP0_ [2:0] |     |     | CD0_[1:0] ID0_ |     |     | [1:0] | RW0_ | [1:0] | SZ0_ | [1:0] |
| Initial value: | 0  | 0  | 0     | 0  | 0  | 0          | 0   | 0   | 0              | 0   | 0   | 0     | 0    | 0     | 0    | 0     |
| R/W:           | R  | R  | R/W   | R  | R  | R/W        | R/W | R/W | R/W            | R/W | R/W | R/W   | R/W  | R/W   | R/W  | R/W   |

| D:4     | Bit Name  | Initial<br>Value | R/W | Description  |
|---------|-----------|------------------|-----|--|
| Bit     | Dit Name  |                  |     | Description  |
| 15, 14  | _         | All 0            | R   | Reserved   |
|         |           |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 13      | UBID0     | 0                | R/W | User Break Interrupt Disable 0   |
|         |           |                  |     | Disables or enables user break interrupt requests when a channel 0 break condition is satisfied.   |
|         |           |                  |     | 0: User break interrupt requests enabled   |
|         |           |                  |     | 1: User break interrupt requests disabled  |
| 12, 11  | _         | All 0            | R   | Reserved   |
|         |           |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 10 to 8 | CP0_[2:0] | 000              | R/W | I-Bus Bus Master Select 0  |
|         |           |                  |     | Select the bus master when the bus cycle of the channel 0 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). |
|         |           |                  |     | xx1: CPU cycle is included in break conditions   |
|         |           |                  |     | x1x: DMAC cycle is included in break conditions  |
|         |           |                  |     | 1xx: A-DMAC cycle is included in break conditions  |

| D.,  | D'AN      | Initial | D 444 | B tota   |
|------|-----------|---------|-------|--|
| Bit  | Bit Name  | Value   | R/W   | Description  |
| 7, 6 | CD0_[1:0] | 00      | R/W   | C Bus Cycle/I Bus Cycle Select 0   |
|      |           |         |       | Select the C bus cycle or I bus cycle as the bus cycle of the channel 0 break condition.   |
|      |           |         |       | 00: Condition comparison is not performed  |
|      |           |         |       | 01: Break condition is the C bus (F bus or M bus) cycle  |
|      |           |         |       | 10: Break condition is the I bus cycle   |
|      |           |         |       | 11: Break condition is the C bus (F bus or M bus) cycle  |
| 5, 4 | ID0_[1:0] | 00      | R/W   | Instruction Fetch/Data Access Select 0   |
|      |           |         |       | Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 0 break condition. If the instruction fetch cycle is selected, select the C bus cycle. |
|      |           |         |       | 00: Condition comparison is not performed  |
|      |           |         |       | 01: Break condition is the instruction fetch cycle   |
|      |           |         |       | 10: Break condition is the data access cycle   |
|      |           |         |       | <ol> <li>Break condition is the instruction fetch cycle or<br/>data access cycle</li> </ol>  |
| 3, 2 | RW0_[1:0] | 00      | R/W   | Read/Write Select 0  |
|      |           |         |       | Select the read cycle or write cycle as the bus cycle of the channel 0 break condition.  |
|      |           |         |       | 00: Condition comparison is not performed  |
|      |           |         |       | 01: Break condition is the read cycle  |
|      |           |         |       | 10: Break condition is the write cycle   |
|      |           |         |       | 11: Break condition is the read cycle or write cycle   |
| 1, 0 | SZ0_[1:0] | 00      | R/W   | Operand Size Select 0  |
|      |           |         |       | Select the operand size of the bus cycle for the channel 0 break condition.  |
|      |           |         |       | 00: Break condition does not include operand size  |
|      |           |         |       | 01: Break condition is byte access   |
|      |           |         |       | 10: Break condition is word access   |
|      |           |         |       | 11: Break condition is longword access   |

[Legend]

x: Don't care

## 9.3.4 Break Address Register\_1 (BAR\_1)

10:4:01

BAR\_1 is a 32-bit readable/writable register. BAR\_1 specifies the address used as a break condition in channel 1. The control bits CD1\_1 and CD1\_0 in the break bus cycle register\_1 (BBR\_1) select one of the two address buses for a channel 1 break condition. BAR\_1 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:                   | 31         | 30         | 29         | 28         | 27         | 26         | 25         | 24         | 23         | 22         | 21         | 20         | 19         | 18         | 17         | 16         |
|------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                        | BA1_<br>31 | BA1_<br>30 | BA1_<br>29 | BA1_<br>28 | BA1_<br>27 | BA1_<br>26 | BA1_<br>25 | BA1_<br>24 | BA1_<br>23 | BA1_<br>22 | BA1_<br>21 | BA1_<br>20 | BA1_<br>19 | BA1_<br>18 | BA1_<br>17 | BA1_<br>16 |
| Initial value:<br>R/W: | 0<br>R/W   |
| Bit:                   | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|                        | BA1_<br>15 | BA1_<br>14 | BA1_<br>13 | BA1_<br>12 | BA1_<br>11 | BA1_<br>10 | BA1_<br>9  | BA1_<br>8  | BA1_<br>7  | BA1_<br>6  | BA1_<br>5  | BA1_<br>4  | BA1_<br>3  | BA1_<br>2  | BA1_<br>1  | BA1_<br>0  |
| Initial value:         | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:                   | R/W        |

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

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR 1 to 0.

## 9.3.5 Break Address Mask Register\_1 (BAMR\_1)

BAMR\_1 is a 32-bit readable/writable register. BAMR\_1 specifies bits masked in the break address bits specified by BAR\_1. BAMR\_1 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 31          | 30          | 29          | 28          | 27          | 26          | 25          | 24          | 23          | 22          | 21          | 20          | 19          | 18          | 17          | 16          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | BAM1_<br>31 | BAM1_<br>30 | BAM1_<br>29 | BAM1_<br>28 | BAM1_<br>27 | BAM1_<br>26 | BAM1_<br>25 | BAM1_<br>24 | BAM1_<br>23 | BAM1_<br>22 | BAM1_<br>21 | BAM1_<br>20 | BAM1_<br>19 | BAM1_<br>18 | BAM1_<br>17 | BAM1_<br>16 |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |
|                |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9           | 8           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|                | BAM1_       | BAM1_       | BAM1_       |             | BAM1_       |             | BAM1_       |             | BAM1_       |             |             |             | BAM1_       |             | BAM1_       | BAM1_       |
|                | 15          | 14          | 13          | 12          | 11          | 10          | 9           | 8           | /           | 6           | 5           | 4           | 3           | 2           |             | 0           |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |

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

# 9.3.6 Break Bus Cycle Register\_1 (BBR\_1)

BBR\_1 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 1. BBR\_1 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 15 | 14 | 13    | 12 | 11 | 10  | 9      | 8   | 7    | 6       | 5    | 4     | 3   | 2       | 1    | 0       |
|----------------|----|----|-------|----|----|-----|--------|-----|------|---------|------|-------|-----|---------|------|---------|
|                | -  | -  | UBID1 | -  | -  | С   | P1_ [2 | :0] | CD1_ | _ [1:0] | ID1_ | [1:0] | RW1 | _ [1:0] | SZ1_ | _ [1:0] |
| Initial value: | 0  | 0  | 0     | 0  | 0  | 0   | 0      | 0   | 0    | 0       | 0    | 0     | 0   | 0       | 0    | 0       |
| R/W:           | R  | R  | R/W   | R  | R  | R/W | R/W    | R/W | R/W  | R/W     | R/W  | R/W   | R/W | R/W     | R/W  | R/W     |

| Bit     | Bit Name  | Initial<br>Value | R/W | Description  |
|---------|-----------|------------------|-----|--|
| 15, 14  | _         | All 0            | R   | Reserved   |
|         |           |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 13      | UBID1     | 0                | R/W | User Break Interrupt Disable 1   |
|         |           |                  |     | Disables or enables user break interrupt requests when a channel 1 break condition is satisfied.   |
|         |           |                  |     | 0: User break interrupt requests enabled   |
|         |           |                  |     | 1: User break interrupt requests disabled  |
| 12, 11  | _         | All 0            | R   | Reserved   |
|         |           |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 10 to 8 | CP1_[2:0] | 000              | R/W | I-Bus Bus Master Select 1  |
|         |           |                  |     | Select the bus master when the bus cycle of the channel 1 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). |
|         |           |                  |     | xx1: CPU cycle is included in break conditions   |
|         |           |                  |     | x1x: DMAC cycle is included in break conditions  |
|         |           |                  |     | 1xx: A-DMAC cycle is included in break conditions  |

Apr 01, 2014

|      |           | Initial |     |  |
|------|-----------|---------|-----|--|
| Bit  | Bit Name  | Value   | R/W | Description  |
| 7, 6 | CD1_[1:0] | 00      | R/W | C Bus Cycle/I Bus Cycle Select 1   |
|      |           |         |     | Select the C bus cycle or I bus cycle as the bus cycle of the channel 1 break condition.   |
|      |           |         |     | 00: Condition comparison is not performed  |
|      |           |         |     | 01: Break condition is the C bus cycle   |
|      |           |         |     | 10: Break condition is the I bus cycle   |
|      |           |         |     | 11: Break condition is the C bus cycle   |
| 5, 4 | ID1_[1:0] | 00      | R/W | Instruction Fetch/Data Access Select 1   |
|      |           |         |     | Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 1 break condition. If the instruction fetch cycle is selected, select the C bus cycle. |
|      |           |         |     | 00: Condition comparison is not performed  |
|      |           |         |     | 01: Break condition is the instruction fetch cycle   |
|      |           |         |     | 10: Break condition is the data access cycle   |
|      |           |         |     | <ol> <li>Break condition is the instruction fetch cycle or<br/>data access cycle</li> </ol>  |
| 3, 2 | RW1_[1:0] | 00      | R/W | Read/Write Select 1  |
|      |           |         |     | Select the read cycle or write cycle as the bus cycle of the channel 1 break condition.  |
|      |           |         |     | 00: Condition comparison is not performed  |
|      |           |         |     | 01: Break condition is the read cycle  |
|      |           |         |     | 10: Break condition is the write cycle   |
|      |           |         |     | 11: Break condition is the read cycle or write cycle   |
| 1, 0 | SZ1_[1:0] | 00      | R/W | Operand Size Select 1  |
|      |           |         |     | Select the operand size of the bus cycle for the channel 1 break condition.  |
|      |           |         |     | 00: Break condition does not include operand size  |
|      |           |         |     | 01: Break condition is byte access   |
|      |           |         |     | 10: Break condition is word access   |
|      |           |         |     | 11: Break condition is longword access   |

[Legend]

x: Don't care

## 9.3.7 Break Address Register\_2 (BAR\_2)

BAR\_2 is a 32-bit readable/writable register. BAR\_2 specifies the address used as a break condition in channel 2. The control bits CD2\_1 and CD2\_0 in the break bus cycle register\_2 (BBR\_2) select one of the two address buses for a channel 2 break condition. BAR\_2 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 31         | 30         | 29         | 28         | 27         | 26         | 25         | 24         | 23         | 22         | 21         | 20         | 19         | 18         | 17         | 16         |
|----------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | BA2_<br>31 | BA2_<br>30 | BA2_<br>29 | BA2_<br>28 | BA2_<br>27 | BA2_<br>26 | BA2_<br>25 | BA2_<br>24 | BA2_<br>23 | BA2_<br>22 | BA2_<br>21 | BA2_<br>20 | BA2_<br>19 | BA2_<br>18 | BA2_<br>17 | BA2_<br>16 |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W        |
|                |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |
| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|                | BA2_<br>15 | BA2_<br>14 | BA2_<br>13 | BA2_<br>12 | BA2_       | BA2_<br>10 | BA2_<br>9  | BA2_<br>8  | BA2_       | BA2_       | BA2_<br>5  | BA2_<br>4  | BA2_<br>3  | BA2_<br>2  | BA2_       | BA2_       |
|                |            | 17         |            | 12         |            | 10         |            |            |            |            |            |            |            |            |            | U          |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W        |

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

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR 2 to 0.

# 9.3.8 Break Address Mask Register\_2 (BAMR\_2)

BAMR\_2 is a 32-bit readable/writable register. BAMR\_2 specifies bits masked in the break address bits specified by BAR\_2. BAMR\_2 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 31          | 30          | 29          | 28          | 27          | 26          | 25          | 24          | 23          | 22          | 21          | 20          | 19          | 18          | 17          | 16          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | BAM2_<br>31 | BAM2_<br>30 | BAM2_<br>29 | BAM2_<br>28 | BAM2_<br>27 | BAM2_<br>26 | BAM2_<br>25 | BAM2_<br>24 | BAM2_<br>23 | BAM2_<br>22 | BAM2_<br>21 | BAM2_<br>20 | BAM2_<br>19 | BAM2_<br>18 | BAM2_<br>17 | BAM2_<br>16 |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |
|                |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9           | 8           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|                | BAM2_<br>15 | BAM2_<br>14 | BAM2_<br>13 | BAM2_<br>12 | BAM2_<br>11 | BAM2_<br>10 | BAM2_<br>9  | BAM2_<br>8  | BAM2_<br>7  | BAM2_       | BAM2_<br>5  | BAM2_<br>4  | BAM2_<br>3  | BAM2_<br>2  | BAM2_<br>1  | BAM2_<br>0  |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |

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

# 9.3.9 Break Bus Cycle Register\_2 (BBR\_2)

BBR\_2 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 2. BBR\_2 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 15 | 14 | 13    | 12 | 11 | 10  | 9         | 8   | 7    | 6     | 5    | 4     | 3    | 2     | 1    | 0     |
|----------------|----|----|-------|----|----|-----|-----------|-----|------|-------|------|-------|------|-------|------|-------|
|                | -  | -  | UBID2 | -  | -  | C   | CP2_ [2:0 | 0]  | CD2_ | [1:0] | ID2_ | [1:0] | RW2_ | [1:0] | SZ2_ | [1:0] |
| Initial value: | 0  | 0  | 0     | 0  | 0  | 0   | 0         | 0   | 0    | 0     | 0    | 0     | 0    | 0     | 0    | 0     |
| R/W:           | R  | R  | R/W   | R  | R  | R/W | R/W       | R/W | R/W  | R/W   | R/W  | R/W   | R/W  | R/W   | R/W  | R/W   |

| Bit     | Bit Name  | Initial<br>Value | R/W | Description  |
|---------|-----------|------------------|-----|--|
| 15, 14  | _         | All 0            | R   | Reserved   |
|         |           |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 13      | UBID2     | 0                | R/W | User Break Interrupt Disable 2   |
|         |           |                  |     | Disables or enables user break interrupt requests when a channel 2 break condition is satisfied.   |
|         |           |                  |     | 0: User break interrupt requests enabled   |
|         |           |                  |     | 1: User break interrupt requests disabled  |
| 12, 11  | _         | All 0            | R   | Reserved   |
|         |           |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 10 to 8 | CP2_[2:0] | 000              | R/W | I-Bus Bus Master Select 2  |
|         |           |                  |     | Select the bus master when the bus cycle of the channel 2 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). |
|         |           |                  |     | xx1: CPU cycle is included in break conditions   |
|         |           |                  |     | x1x: DMAC cycle is included in break conditions  |
|         |           |                  |     | 1xx: A-DMAC cycle is included in break conditions  |

| Bit  | Bit Name  | Initial<br>Value | R/W | Description  |
|------|-----------|------------------|-----|--|
| 7, 6 | CD2_[1:0] | 00               | R/W | C Bus Cycle/I Bus Cycle Select 2   |
|      |           |                  |     | Select the C bus cycle or I bus cycle as the bus cycle of the channel 2 break condition.   |
|      |           |                  |     | 00: Condition comparison is not performed  |
|      |           |                  |     | 01: Break condition is the C bus cycle   |
|      |           |                  |     | 10: Break condition is the I bus cycle   |
|      |           |                  |     | 11: Break condition is the C bus cycle   |
| 5, 4 | ID2_[1:0] | 00               | R/W | Instruction Fetch/Data Access Select 2   |
|      |           |                  |     | Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 2 break condition. If the instruction fetch cycle is selected, select the C bus cycle. |
|      |           |                  |     | 00: Condition comparison is not performed  |
|      |           |                  |     | 01: Break condition is the instruction fetch cycle   |
|      |           |                  |     | 10: Break condition is the data access cycle   |
|      |           |                  |     | <ol> <li>Break condition is the instruction fetch cycle or<br/>data access cycle</li> </ol>  |
| 3, 2 | RW2_[1:0] | 00               | R/W | Read/Write Select 2  |
|      |           |                  |     | Select the read cycle or write cycle as the bus cycle of the channel 2 break condition.  |
|      |           |                  |     | 00: Condition comparison is not performed  |
|      |           |                  |     | 01: Break condition is the read cycle  |
|      |           |                  |     | 10: Break condition is the write cycle   |
|      |           |                  |     | 11: Break condition is the read cycle or write cycle   |
| 1, 0 | SZ2_[1:0] | 00               | R/W | Operand Size Select 2  |
|      |           |                  |     | Select the operand size of the bus cycle for the channel 2 break condition.  |
|      |           |                  |     | 00: Break condition does not include operand size  |
|      |           |                  |     | 01: Break condition is byte access   |
|      |           |                  |     | 10: Break condition is word access   |
|      |           |                  |     | 11: Break condition is longword access   |

[Legend]

x: Don't care

## 9.3.10 Break Address Register\_3 (BAR\_3)

BAR\_3 is a 32-bit readable/writable register. BAR\_3 specifies the address used as a break condition in channel 3. The control bits CD3\_1 and CD3\_0 in the break bus cycle register\_3 (BBR\_3) select one of the two address buses for a channel 3 break condition. BAR\_3 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 31         | 30         | 29         | 28         | 27         | 26         | 25         | 24         | 23         | 22         | 21         | 20         | 19         | 18         | 17         | 16         |
|----------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | BA3_<br>31 | BA3_<br>30 | BA3_<br>29 | BA3_<br>28 | BA3_<br>27 | BA3_<br>26 | BA3_<br>25 | BA3_<br>24 | BA3_<br>23 | BA3_<br>22 | BA3_<br>21 | BA3_<br>20 | BA3_<br>19 | BA3_<br>18 | BA3_<br>17 | BA3_<br>16 |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W        |
|                |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |
| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|                | BA3_       |
|                | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8          | /          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W        |

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

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR 3 to 0.

# 9.3.11 Break Address Mask Register\_3 (BAMR\_3)

BAMR\_3 is a 32-bit readable/writable register. BAMR\_3 specifies bits masked in the break address bits specified by BAR\_3. BAMR\_3 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 31          | 30          | 29          | 28          | 27          | 26          | 25          | 24          | 23          | 22          | 21          | 20          | 19          | 18          | 17          | 16          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | BAM3_<br>31 | BAM3_<br>30 | BAM3_<br>29 | BAM3_<br>28 | BAM3_<br>27 | BAM3_<br>26 | BAM3_<br>25 | BAM3_<br>24 | BAM3_<br>23 | BAM3_<br>22 | BAM3_<br>21 | BAM3_<br>20 | BAM3_<br>19 | BAM3_<br>18 | BAM3_<br>17 | BAM3_<br>16 |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |
|                |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9           | 8           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|                | BAM3_<br>15 | BAM3_<br>14 | BAM3_<br>13 | BAM3_<br>12 | BAM3_<br>11 | BAM3_<br>10 | BAM3_<br>9  | BAM3_<br>8  | BAM3_<br>7  | BAM3_       | BAM3_<br>5  | BAM3_<br>4  | BAM3_       | BAM3_       | BAM3_<br>1  | BAM3_<br>0  |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |

|         |            | Initial |     |   |
|---------|------------|---------|-----|---|
| Bit     | Bit Name   | Value   | R/W | Description   |
| 31 to 0 | BAM3_31 to | All 0   | R/W | Break Address Mask 3  |
|         | BAM3_0     |         |     | Specify bits masked in the channel 3 break address bits specified by BAR_3 (BA3_31 to BA3_0).     |
|         |            |         |     | Break address bit BA3_n is included in the break condition  |
|         |            |         |     | <ol> <li>Break address bit BA3_n is masked and not<br/>included in the break condition</li> </ol> |
|         |            |         |     | Note: $n = 31$ to 0   |

#### Break Bus Cycle Register\_3 (BBR\_3) 9.3.12

BBR\_3 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 3. BBR\_3 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 15 | 14 | 13    | 12 | 11 | 10  | 9        | 8   | 7    | 6     | 5    | 4     | 3    | 2     | 1    | 0     |
|----------------|----|----|-------|----|----|-----|----------|-----|------|-------|------|-------|------|-------|------|-------|
|                | -  | -  | UBID3 | -  | -  | С   | P3_ [2:0 | )]  | CD3_ | [1:0] | ID3_ | [1:0] | RW3_ | [1:0] | SZ3_ | [1:0] |
| Initial value: | 0  | 0  | 0     | 0  | 0  | 0   | 0        | 0   | 0    | 0     | 0    | 0     | 0    | 0     | 0    | 0     |
| R/W:           | R  | R  | R/W   | R  | R  | R/W | R/W      | R/W | R/W  | R/W   | R/W  | R/W   | R/W  | R/W   | R/W  | R/W   |

| Bit     | Bit Name  | Initial<br>Value | R/W | Description  |
|---------|-----------|------------------|-----|--|
| 15, 14  | _         | All 0            | R   | Reserved   |
|         |           |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 13      | UBID3     | 0                | R/W | User Break Interrupt Disable 3   |
|         |           |                  |     | Disables or enables user break interrupt requests when a channel 3 break condition is satisfied.   |
|         |           |                  |     | 0: User break interrupt requests enabled   |
|         |           |                  |     | 1: User break interrupt requests disabled  |
| 12, 11  | _         | All 0            | R   | Reserved   |
|         |           |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 10 to 8 | CP3_[2:0] | 000              | R/W | I-Bus Bus Master Select 3  |
|         |           |                  |     | Select the bus master when the bus cycle of the channel 3 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). |
|         |           |                  |     | xx1: CPU cycle is included in break conditions   |
|         |           |                  |     | x1x: DMAC cycle is included in break conditions  |
|         |           |                  |     | 1xx: A-DMAC cycle is included in break conditions  |

| Bit  | Bit Name  | Initial<br>Value | R/W | Description  |
|------|-----------|------------------|-----|--|
| 7, 6 | CD3_[1:0] | 00               | R/W | C Bus Cycle/I Bus Cycle Select 3   |
|      |           |                  |     | Select the C bus cycle or I bus cycle as the bus cycle of the channel 3 break condition.   |
|      |           |                  |     | 00: Condition comparison is not performed  |
|      |           |                  |     | 01: Break condition is the C bus cycle   |
|      |           |                  |     | 10: Break condition is the I bus cycle   |
|      |           |                  |     | 11: Break condition is the C bus cycle   |
| 5, 4 | ID3_[1:0] | 00               | R/W | Instruction Fetch/Data Access Select 3   |
|      |           |                  |     | Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 3 break condition. If the instruction fetch cycle is selected, select the C bus cycle. |
|      |           |                  |     | 00: Condition comparison is not performed  |
|      |           |                  |     | 01: Break condition is the instruction fetch cycle   |
|      |           |                  |     | 10: Break condition is the data access cycle   |
|      |           |                  |     | <ol> <li>Break condition is the instruction fetch cycle or<br/>data access cycle</li> </ol>  |
| 3, 2 | RW3_[1:0] | 00               | R/W | Read/Write Select 3  |
|      |           |                  |     | Select the read cycle or write cycle as the bus cycle of the channel 3 break condition.  |
|      |           |                  |     | 00: Condition comparison is not performed  |
|      |           |                  |     | 01: Break condition is the read cycle  |
|      |           |                  |     | 10: Break condition is the write cycle   |
|      |           |                  |     | 11: Break condition is the read cycle or write cycle   |
| 1, 0 | SZ3_[1:0] | 00               | R/W | Operand Size Select 3  |
|      |           |                  |     | Select the operand size of the bus cycle for the channel 3 break condition.  |
|      |           |                  |     | 00: Break condition does not include operand size  |
|      |           |                  |     | 01: Break condition is byte access   |
|      |           |                  |     | 10: Break condition is word access   |
|      |           |                  |     | 11: Break condition is longword access   |

[Legend]

x: Don't care

#### **Break Control Register (BRCR)** 9.3.13

BRCR sets the following conditions:

- 1. Specifies whether a break is set before or after instruction execution.
- 2. Specifies the pulse width of the UBCTRG output when a break condition is satisfied.
- 3. Specifies whether or not the UBCTRG output is disabled when a break condition is satisfied.

BRCR is a 32-bit readable/writable register that has break condition match flags and bits for setting other break conditions. For the condition match flags of bits 15 to 8, writing 1 is invalid (previous values are retained) and writing 0 is only possible. To clear the flag, write 0 to the flag bit to be cleared and 1 to all other flag bits. BRCR is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

| Bit:           | 31         | 30         | 29         | 28         | 27         | 26         | 25         | 24         | 23   | 22   | 21    | 20    | 19    | 18    | 17  | 16     |
|----------------|------------|------------|------------|------------|------------|------------|------------|------------|------|------|-------|-------|-------|-------|-----|--------|
|                | -          | -          | -          | -          | -          | -          | -          | -          | -    | -    | UTOD3 | UTOD2 | UTOD1 | UTOD0 | CKS | S[1:0] |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0    | 0    | 0     | 0     | 0     | 0     | 0   | 0      |
| R/W:           | R          | R          | R          | R          | R          | R          | R          | R          | R    | R    | R/W   | R/W   | R/W   | R/W   | R/W | R/W    |
| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8          | 7    | 6    | 5     | 4     | 3     | 2     | 1   | 0      |
|                | SCMFC<br>0 | SCMFC<br>1 | SCMFC<br>2 | SCMFC<br>3 | SCMFD<br>0 | SCMFD<br>1 | SCMFD<br>2 | SCMFD<br>3 | PCB3 | PCB2 | PCB1  | PCB0  | 1     | -     | -   | -      |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0    | 0    | 0     | 0     | 0     | 0     | 0   | 0      |
| R/W:           | R/W        | R/W  | R/W  | R/W   | R/W   | R     | R     | R   | R      |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 31 to 22 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.                                 |
| 21       | UTOD3    | 0                | R/W | UBCTRG Output Disable 3  |
|          |          |                  |     | Specifies whether or not $\overline{UBCTRG}$ output is disabled when a break condition is satisfied. |
|          |          |                  |     | 0: UBCTRG output is not disabled when a break condition for channel 3 is satisfied.                  |
|          |          |                  |     | 1: UBCTRG output is disabled when a break condition for channel 3 is satisfied.                      |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 20     | UTOD2    | 0                | R/W | UBCTRG Output Disable 2   |
|        |          |                  |     | Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied.   |
|        |          |                  |     | 0: UBCTRG output is not disabled when a break condition for channel 2 is satisfied.   |
|        |          |                  |     | 1: UBCTRG output is disabled when a break condition for channel 2 is satisfied.   |
| 19     | UTOD1    | 0                | R/W | UBCTRG Output Disable 1   |
|        |          |                  |     | Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied.   |
|        |          |                  |     | 0: UBCTRG output is not disabled when a break condition for channel 1 is satisfied.   |
|        |          |                  |     | 1: UBCTRG output is disabled when a break condition for channel 1 is satisfied.   |
| 18     | UTOD0    | 0                | R/W | UBCTRG Output Disable 0   |
|        |          |                  |     | Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied.   |
|        |          |                  |     | 0: UBCTRG output is not disabled when a break condition for channel 0 is satisfied.   |
|        |          |                  |     | 1: UBCTRG output is disabled when a break condition for channel 0 is satisfied.   |
| 17, 16 | CKS[1:0] | 00               | R/W | Clock Select  |
|        |          |                  |     | Specifies the pulse width output to the $\overline{\text{UBCTRG}}$ pin when a break condition is satisfied.   |
|        |          |                  |     | 00: Pulse width of $\overline{\text{UBCTRG}}$ is one P $_{\varphi}$ cycle (prohibited when two-time multiplication peripheral clock is set)                     |
|        |          |                  |     | 01: Pulse width of UBCTRG is two Pφ cycles  |
|        |          |                  |     | 10: Pulse width of $\overline{UBCTRG}$ is four $P\phi$ cycles   |
|        |          |                  |     | 11: Pulse width of $\overline{\text{UBCTRG}}$ is eight P $\phi$ cycles  |
| 15     | SCMFC0   | 0                | R/W | C Bus Cycle Condition Match Flag 0  |
|        |          |                  |     | When the C bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. |
|        |          |                  |     | 0: The C bus cycle condition for channel 0 does not match   |
|        |          |                  |     | 1: The C bus cycle condition for channel 0 matches  |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 14  | SCMFC1   | 0                | R/W | C Bus Cycle Condition Match Flag 1  |
|     |          |                  |     | When the C bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. |
|     |          |                  |     | 0: The C bus cycle condition for channel 1 does not match   |
|     |          |                  |     | 1: The C bus cycle condition for channel 1 matches  |
| 13  | SCMFC2   | 0                | R/W | C Bus Cycle Condition Match Flag 2  |
|     |          |                  |     | When the C bus cycle condition in the break conditions set for channel 2 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. |
|     |          |                  |     | 0: The C bus cycle condition for channel 2 does not match   |
|     |          |                  |     | 1: The C bus cycle condition for channel 2 matches  |
| 12  | SCMFC3   | 0                | R/W | C Bus Cycle Condition Match Flag 3  |
|     |          |                  |     | When the C bus cycle condition in the break conditions set for channel 3 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. |
|     |          |                  |     | 0: The C bus cycle condition for channel 3 does not match   |
|     |          |                  |     | 1: The C bus cycle condition for channel 3 matches  |
| 11  | SCMFD0   | 0                | R/W | I Bus Cycle Condition Match Flag 0  |
|     |          |                  |     | When the I bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. |
|     |          |                  |     | 0: The I bus cycle condition for channel 0 does not match   |
|     |          |                  |     | 1: The I bus cycle condition for channel 0 matches  |
| 10  | SCMFD1   | 0                | R/W | I Bus Cycle Condition Match Flag 1  |
|     |          |                  |     | When the I bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. |
|     |          |                  |     | 0: The I bus cycle condition for channel 1 does not match   |
|     |          |                  |     | 1: The I bus cycle condition for channel 1 matches  |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 9   | SCMFD2   | 0                | R/W | I Bus Cycle Condition Match Flag 2  |
|     |          |                  |     | When the I bus cycle condition in the break conditions set for channel 2 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. |
|     |          |                  |     | The I bus cycle condition for channel 2 does not match  |
|     |          |                  |     | 1: The I bus cycle condition for channel 2 matches  |
| 8   | SCMFD3   | 0                | R/W | I Bus Cycle Condition Match Flag 3  |
|     |          |                  |     | When the I bus cycle condition in the break conditions set for channel 3 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. |
|     |          |                  |     | 0: The I bus cycle condition for channel 3 does not match   |
|     |          |                  |     | 1: The I bus cycle condition for channel 3 matches  |
| 7   | PCB3     | 0                | R/W | PC Break Select 3   |
|     |          |                  |     | Selects the break timing of the instruction fetch cycle for channel 3 as before or after instruction execution.   |
|     |          |                  |     | <ol> <li>PC break of channel 3 is generated before<br/>instruction execution</li> </ol>   |
|     |          |                  |     | 1: PC break of channel 3 is generated after instruction execution   |
| 6   | PCB2     | 0                | R/W | PC Break Select 2   |
|     |          |                  |     | Selects the break timing of the instruction fetch cycle for channel 2 as before or after instruction execution.   |
|     |          |                  |     | PC break of channel 2 is generated before instruction execution   |
|     |          |                  |     | 1: PC break of channel 2 is generated after instruction execution   |
| 5   | PCB1     | 0                | R/W | PC Break Select 1   |
|     |          |                  |     | Selects the break timing of the instruction fetch cycle for channel 1 as before or after instruction execution.   |
|     |          |                  |     | PC break of channel 1 is generated before instruction execution   |
|     |          |                  |     | 1: PC break of channel 1 is generated after instruction execution   |

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

# 9.4 Operation

#### 9.4.1 Flow of the User Break Operation

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

- 1. The break address is set in a break address register (BAR). The masked address bits are set in a break address mask register (BAMR). The bus break conditions are set in the break bus cycle register (BBR). Three control bit groups of BBR (C bus cycle/I bus cycle select, instruction fetch/data access select, and read/write select) are each set. No user break will be generated if even one of these groups is set to 00. The relevant break control conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBR, and branch after reading from the last written register and executing more than five NOP instructions. The newly written register values become valid from the instruction at the branch destination.
- 2. When the break conditions are satisfied, the UBC sends a user break request to the CPU, sets the C bus condition match flag (SCMFC) or I bus condition match flag (SCMFD) for the appropriate channel, and outputs a pulse to the UBCTRG pin with the width set by the CKS[1:0] bits. Setting the UBID bit in BBR to 1 enables external monitoring of the trigger output without requesting user break interrupts.
- 3. On receiving a user break interrupt request signal, the INTC determines its priority. Since the user break interrupt has a priority level of 15, it is accepted when the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR) is 14 or lower. If the I3 to I0 bits are set to a priority level of 15, the user break interrupt is not accepted, but the conditions are checked, and condition match flags are set if the conditions match. For details on ascertaining the priority, see section 8, Interrupt Controller (INTC).
- 4. Condition match flags (SCMFC and SCMFD) can be used to check which condition has been satisfied. The flag is set on a condition match and is not cleared automatically. Before using the flags again, 0 must first be written to them. The condition match flags must be cleared during the user break interrupt exception handling routine. Otherwise, an interrupt will be generated again.
- 5. There is a chance that the break set in channel 0 and the break set in channel 1 occur around the same time. In this case, there will be only one break request to the CPU, but these two break channel match flags may both be set.
- 6. When selecting the I bus as the break condition, note as follows:
  - Several bus masters, including the CPU and DMAC, are connected to the I bus. The UBC
    monitors bus cycles generated by the bus master specified by BBR, and determines the
    condition match.

- I bus cycles resulting from instruction fetches on the C bus by the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles are defined as data access cycles.
- I bus cycles the DMAC or A-DMAC issues are only data access cycles.
- If a break condition is specified for the I bus, even when the condition matches in an I bus cycle resulting from an instruction executed by the CPU, at which instruction the break is to be accepted cannot be clearly defined.

#### 9.4.2 **Break on Instruction Fetch Cycle**

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

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

- 3. When setting a break condition for break after instruction execution, the instruction set with the break condition is executed and then the break is generated prior to execution of the next instruction. As with pre-execution breaks, this function is not available for overrun fetch instructions. When this kind of break is set for a delayed branch instruction and its delay slot, a break is not generated until the first instruction at the branch destination.
- 4. If the I bus is set for a break of an instruction fetch cycle, the setting is invalidated.

### 9.4.3 Break on Data Access Cycle

- 1. If the C bus is specified as a break condition for data access break, condition comparison is performed for the logical addresses accessed by the executed instructions, and a break occurs if the condition is satisfied. If the I bus is specified as a break condition, condition comparison is performed for the physical addresses of the data access cycles that are issued by the bus master specified by the bits to select the bus master of the I bus, and a break occurs if the condition is satisfied. For details on the CPU bus cycles issued on the I bus, see 6 in section 9.4.1, Flow of the User Break Operation.
- 2. The relationship between the data access cycle address and the comparison condition for each operand size is listed in table 9.3.

Table 9.3 Data Access Cycle Addresses and Operand Size Comparison Conditions

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

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

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

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

### 9.4.4 Value of Saved Program Counter

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

- 1. When C bus (FAB)/instruction fetch (before instruction execution) is specified as a break condition:
  - The address of the instruction that matched the break condition is saved to the stack. The instruction that matched the condition is not executed, and the break occurs before it. However when a delay slot instruction matches the condition, the instruction is executed, and the branch destination address is saved to the stack.
- 2. When C bus (FAB)/instruction fetch (after instruction execution) is specified as a break condition:
  - The address of the instruction following the instruction that matched the break condition is saved to the stack. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delayed branch instruction or delay slot matches the condition, the instruction is executed, and the branch destination address is saved to the stack.
- When C bus/data access or I bus/data access is specified as a break condition:
   The address after executing several instructions of the instruction that matched the break condition is saved to the stack.

#### 9.4.5 Usage Examples

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

(Example 1-1)

• Register specifications

BAR\_0 = H'00000404, BAMR\_0 = H'00000000, BBR\_0 = H'0054, BAR\_1 = H'00008010, BAMR\_1 = H'00000006, BBR\_1 = H'0054, BRCR = H'00000010

<Channel 0>

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

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

<Channel 1>

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

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

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

included in the condition)

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

(Example 1-2)

• Register specifications

BAR\_0 = H'00027128, BAMR\_0 = H'00000000, BBR\_0 = H'005A, BAR\_1 = H'00031415, BAMR\_1 = H'00000000, BBR\_1 = H'0054, BRCR = H'00000000

<Channel 0>

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

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

<Channel 1>

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

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

included in the condition)

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

#### (Example 1-3)

#### • Register specifications

BAR\_0 = H'00008404, BAMR\_0 = H'00000FFF, BBR\_0 = H'0054, BAR\_1 = H'00008010, BAMR\_1 = H'00000006, BBR\_1 = H'0054, BRCR = H'00000010

<Channel 0>

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

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

included in the condition)

<Channel 1>

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

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

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

included in the condition)

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

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

(Example 2-1)

### Register specifications

BAR\_0 = H'00123456, BAMR\_0 = H'00000000, BBR\_0 = H'0064, BAR\_1 = H'000ABCDE, BAMR\_1 = H'000000FF, BBR\_1 = H'006A, BRCR = H'00000000

<Channel 0>

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

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

<Channel 1>

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

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

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

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

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

(Example 3-1)

#### • Register specifications

BAR\_0 = H'00314156, BAMR\_0 = H'00000000, BBR\_0 = H'0094, BAR\_1 = H'00055555,

BAMR\_1 = H'00000000, BBR\_1 = H'02A9, BRCR = H'00000000

<Channel 0>

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

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

<Channel 1>

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

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

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

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

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

# 9.5 Usage Notes

- The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the
  period from executing an instruction to rewrite the UBC register till the new value is actually
  rewritten, the desired break may not occur. In order to know the timing when the UBC register
  is changed, execute five or more NOP instructions after reading from the last written register.
  Instructions after then are valid for the newly written register value.
- 2. The UBC cannot monitor access to the C bus and I bus cycles in the same channel.
- 3. When a user break and another exception occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 7.1 in section 7, Exception Handling. If an exception with higher priority occurs, the user break is not generated.
- 4. Note the following when a break occurs in a delay slot.

  If a pre-execution break is set at a delay slot instruction, the break occurs immediately before execution of the branch destination.
- 5. User breaks are disabled during UBC module standby mode. Do not read from or write to the UBC registers during UBC module standby mode; the values are not guaranteed.
- 6. Do not set an address within an interrupt exception handling routine whose interrupt priority level is at least 15 (including user break interrupts) as a break address.
- 7. Do not set a break after instruction execution for the SLEEP instruction or for the delayed branch instruction where the SLEEP instruction is placed at its delay slot.
- 8. When setting a break for a 32-bit instruction, set the address where the upper 16 bits are placed. If the address of the lower 16 bits is set and a break before instruction execution is set as a break condition, the break is handled as a break after instruction execution.
- 9. Do not place a pre-execution break for the instruction immediately after a DIVU (or DIVS) instruction. Otherwise the break will be still taken even though an exception or interrupt occurs during the DIVU (or DIVS) instruction execution.

# Section 10 Bus State Controller (BSC)

The bus state controller (BSC) outputs control signals for external memory and external devices that are connected to the external address space. BSC functions enable this LSI to connect directly with SRAM, ROM, and other memories such as SRAM with byte selection, and external devices.

#### 10.1 **Features**

- 1. External address space
  - A maximum 4-Mbyte linear space access for each of spaces CS0 to CS3.
  - Can select the data bus width (8 or 16 bits) for each address space.
  - Controls insertion of wait cycles for each address space.
  - Controls insertion of wait cycles for each read access and write access.
  - Can set independent idle cycles during the continuous access for five cases: read-write (in same space/different spaces), read-read (in same space/different spaces), the first cycle is a write access.

Figure 10.1 shows a block diagram of the BSC.

Page 210 of 1824

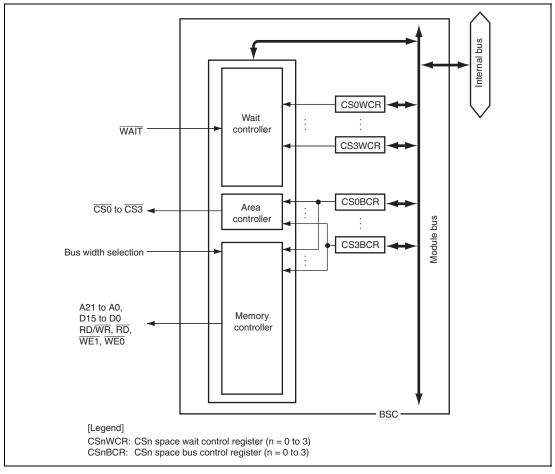


Figure 10.1 Block Diagram of BSC

# 10.2 Input/Output Pins

Table 10.1 shows the pin configuration of the BSC.

**Table 10.1 Pin Configuration** 

| Name                | I/O    | Function  |
|---------------------|--------|---|
| A21 to A0           | Output | External address bus  |
| D15 to D0           | I/O    | External data bus   |
| CS0 to CS3          | Output | Chip select   |
| RD/WR               | Output | Read/write  |
|                     |        | Can be connected to $\overline{\text{WE}}$ pins when SRAM with byte selection is connected. |
| RD                  | Output | Read pulse signal (read data output enable signal)  |
| WE1                 | Output | Indicates that D15 to D8 are being written to.  |
|                     |        | Indicates the byte select signal when a SRAM with byte selection is connected.              |
| WE0                 | Output | Indicates that D7 to D0 are being written to.   |
|                     |        | Indicates the byte select signal when a SRAM with byte selection is connected.              |
| WAIT                | Input  | External wait input   |
| Bus width selection | Input  | Select the initial bus width of address spaces CS0 to CS3.                                  |

#### 10.3 Area Overview

#### 10.3.1 Address Map

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

Any memory space of CS0 to CS3 can be activated by SRAM interface or SRAM interface with byte selection. However, the space CS0 in on-chip ROM disabled mode should be activated by SRAM interface.

The address map for the external address space is listed in tables 10.2 and 10.3.

Table 10.2 Address Map in On-Chip ROM Enabled Mode

| Address   | Type of Address Space   | Capacity                   |
|---|---|----------------------------|
| H'0000 0000 to H'003B FFFF<br>(H'0000 0000 to H'0000 7FFF) *1 | On-chip ROM (user MAT read)<br>(On-chip ROM (user boot MAT read))   | 3.75 Mbytes<br>(32 Kbytes) |
| H'003C 0000 to H'0040 1FFF                                    | Reserved  |                            |
| H'0040 2000 to H'0040 3FFF                                    | FCU firmware area   | 8 Kbytes                   |
| H'0040 4000 to H'01FF FFFF                                    | Reserved  |                            |
| H'0200 0000 to H'023F FFFF                                    | External address space (CS0)  | 4 Mbytes                   |
| H'0240 0000 to H'03FF FFFF                                    | CS0 shadow space  | 28 Mbytes                  |
| H'0400 0000 to H'043F FFFF                                    | External address space (CS1)  | 4 Mbytes                   |
| H'0440 0000 to H'07FF FFFF                                    | CS1 shadow space  | 60 Mbytes                  |
| H'0800 0000 to H'083F FFFF                                    | External address space (CS2)  | 4 Mbytes                   |
| H'0840 0000 to H'0BFF FFFF                                    | CS2 shadow space  | 60 Mbytes                  |
| H'0C00 0000 to H'0C3F FFFF                                    | External address space (CS3)  | 4 Mbytes                   |
| H'0C40 0000 to H'0FFF FFFF                                    | CS3 shadow space  | 60 Mbytes                  |
| H'1000 0000 to H'800F FFFF                                    | Reserved  |                            |
| H'8010 0000 to H'8011 FFFF                                    | EEPROM (read/write)*2   | 128 Kbytes                 |
| H'8012 0000 to H'807F FFFF                                    | Reserved  |                            |
| H'8080 0000 to H'80BB FFFF<br>(H'8080 0000 to H'8080 7FFF) *1 | On-chip ROM (user MAT write)<br>(On-chip ROM (user boot MAT write)) | 3.75 Mbytes<br>(32 Kbytes) |
| H'80BC 0000 to H'80FF 7FFF                                    | Reserved  |                            |
| H'80FF 8000 to H'80FF 9FFF                                    | FCU RAM area  | 8 Kbytes                   |

| Address                    | Type of Address Space               | Capacity             |
|----------------------------|-------------------------------------|----------------------|
| H'80FF A000 to H'FFF7 FFFF | Reserved                            |                      |
| H'FFF8 0000 to H'FFFB FFFF | On-chip RAM* <sup>3</sup>           | 256 Kbytes           |
| H'FFFC 0000 to H'FFFF FFFF | On-chip I/O registers* <sup>3</sup> | 256 Kbytes<br>(Max.) |

- Notes: 1. These addresses apply when the user boot MAT in the on-chip ROM is selected. See section 25, ROM, for details.
  - The EEPROM area includes the lot trace information. See section 26, EEPROM, for details.
  - 3. For the on-chip RAM space, access the addresses shown in section 5, Address Space. For the on-chip I/O register space, access the addresses shown in section 31, List of Registers. Do not access addresses that are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

Table 10.3 Address Map in On-Chip ROM Disabled Mode

| Address                    | Type of Address Space        | Capacity             |
|----------------------------|------------------------------|----------------------|
| H'0000 0000 to H'003F FFFF | External address space (CS0) | 4 Mbytes             |
| H'0040 0000 to H'03FF FFFF | CS0 shadow space             | 60 Mbytes            |
| H'0400 0000 to H'043F FFFF | External address space (CS1) | 4 Mbytes             |
| H'0440 0000 to H'07FF FFFF | CS1 shadow space             | 60 Mbytes            |
| H'0800 0000 to H'083F FFFF | External address space (CS2) | 4 Mbytes             |
| H'0840 0000 to H'0BFF FFFF | CS2 shadow space             | 60 Mbytes            |
| H'0C00 0000 to H'0C3F FFFF | External address space (CS3) | 4 Mbytes             |
| H'0C40 0000 to H'0FFF FFFF | CS3 shadow space             | 60 Mbytes            |
| H'1000 0000 to H'FFFF 7FFF | Reserved                     |                      |
| H'FFF8 0000 to H'FFFB FFFF | On-chip RAM*                 | 256 Kbytes           |
| H'FFFC 0000 to H'FFFF FFFF | On-chip I/O registers*       | 256 Kbytes<br>(Max.) |

Note: \* For the on-chip RAM space, access the addresses shown in section 5, Address Space. For the on-chip I/O register space, access the addresses shown in section 31, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

### 10.3.2 Operating Modes and Data Bus Width in CS0 Space

This LSI can specify the following operating modes at a power-on reset by using the mode setting pins.

### (1) Single Chip Mode/External Address Space Access Enabled Mode

In single chip mode, the external address spaces are not accessed. The addresses, data, and control pins to be used in external address space access enabled mode can be used as ports.

### (2) On-Chip ROM Enabled Mode/On-Chip ROM Disabled Mode

In on-chip ROM enabled mode, first half of the CS0 is allocated in the on-chip ROM. Thus, after a power-on reset, the LSI can be activated by the on-chip ROM program.

In on-chip ROM disabled mode, the LSI is activated by the program stored in the external memory allocated in the CS0. In this case, the external memory in the CS0 functions as the SRAM interface, therefore, the SRAM with byte selection cannot be connected.

To use the RD/ $\overline{WR}$  signal or  $\overline{CS1}$  to  $\overline{CS3}$  signals, these pins must be set by the pin function controller. For details, see section 22, Pin Function Controller (PFC). Execute nothing but accessing CS0 space before a program completes pin function settings.

# (3) Data Bus Width of the Space CS0

In on-chip ROM disabled mode, the data bus width of CS0 area can be set to 8 bits or 16 bits. The data bus widths of CS1 to CS3 areas are specified through register settings. In on-chip ROM enabled mode, all of the data bus widths of the spaces CS0 to CS3 are specified through register settings.

For details on mode settings, see section 3, Operating Modes.

# 10.4 Register Descriptions

The BSC has the following registers.

For addresses and access sizes of these registers, see section 31, List of Registers.

In on-chip ROM disabled mode, do not access spaces other than CS0 area until settings of the connected memory interface are completed.

**Table 10.4 Register Configuration** 

| Register Name                   | Abbreviation | R/W | Initial Value | Address                     | Access<br>Size |
|---------------------------------|--------------|-----|---------------|-----------------------------|----------------|
| CSn space bus control           | CSnBCR       | R/W | H'36DB0400*   | H'FFFC0004 to               | 32             |
| register                        |              |     | H'36DB0200*   | H'FFFC0010                  |                |
| CSn space wait control register | CSnWCR       | R/W | H'00000500    | H'FFFC0028 to<br>H'FFFC0034 | 32             |

Note: \* This initial value is different between in on-chip ROM enabled mode and on-chip ROM disabled mode. In on-chip ROM disabled mode, the initial value varies depending on the settings of the mode pins. For details, see section 10.4.1, CSn Space Bus Control Register (CSnBCR) (n = 0 to 3).

### 10.4.1 CSn Space Bus Control Register (CSnBCR) (n = 0 to 3)

CSnBCR is a 32-bit readable/writable register that specifies the function of each area, the number of idle cycles between bus cycles, and the bus width. In on-chip ROM enabled mode, CSnBCR is initialized to H'36DB0400 at a power-on reset. In on-chip ROM disabled mode, it is initialized to H'36DB0400 (16-bit bus width) or H'36DB0200 (8-bit bus width) depending on the settings of the mode pins.

Do not access external memory other than CS0 before CSnBCR initial setting is completed.

Idle cycles may be inserted even when they are not specified. For details, see section 10.5.6, Wait between Access Cycles.

| Bit:                   | 31     | 30       | 29       | 28       | 27       | 26       | 25             | 24       | 23       | 22       | 21       | 20       | 19       | 18       | 17       | 16       |
|------------------------|--------|----------|----------|----------|----------|----------|----------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                        | -      |          | IWW[2:0] |          | IV       | VRWD[2:  | :0]            | IV       | VRWS[2:  | 0]       | I۱       | VRRD[2:  | 0]       | IV       | VRRS[2:  | 0]       |
| Initial value:<br>R/W: | 0<br>R | 0<br>R/W | 1<br>R/W | 1<br>R/W | 0<br>R/W | 1<br>R/W | 1<br>R/W       | 0<br>R/W | 1<br>R/W | 1<br>R/W | 0<br>R/W | 1<br>R/W | 1<br>R/W | 0<br>R/W | 1<br>R/W | 1<br>R/W |
| Bit:                   | 15     | 14       | 13       | 12       | 11       | 10       | 9              | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|                        | -      | -        | TYPI     | E[1:0]   | -        | BSZ      | <b>[</b> [1:0] | -        | -        | -        | -        | 1        | -        | -        | -        | -        |
| Initial value:         | 0      | 0        | 0        | 0        | 0        | 1/0      | 0/1            | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |
| R/W:                   | R      | R        | R/W      | R/W      | R        | R/W      | R/W            | R        | R        | R        | R        | R        | R        | R        | R        | R        |

|          |          | Initial |     |   |
|----------|----------|---------|-----|---|
| Bit      | Bit Name | Value   | R/W | Description   |
| 31       | _        | 0       | R   | Reserved  |
|          |          |         |     | This bit is always read as 0. The write value should always be 0.   |
| 30 to 28 | IWW[2:0] | 011     | R/W | Idle Cycles between Write-Read Cycles and Write-Write Cycles  |
|          |          |         |     | These bits specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycles are the write-read cycle and write-write cycle. |
|          |          |         |     | 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock)  |
|          |          |         |     | 001: 1 idle cycle inserted (setting prohibited when two-<br>times multiplication has been set for the<br>peripheral clock)  |
|          |          |         |     | 010: 2 idle cycles inserted   |
|          |          |         |     | 011: 4 idle cycles inserted   |
|          |          |         |     | 100: 6 idle cycles inserted   |
|          |          |         |     | 101: 8 idle cycles inserted   |
|          |          |         |     | 110: 10 idle cycles inserted  |
|          |          |         |     | 111: 12 idle cycles inserted  |

| Bit      | Bit Name   | Initial<br>Value | R/W | Description  |
|----------|------------|------------------|-----|--|
| 27 to 25 | IWRWD[2:0] | 011              | R/W | Idle Cycles for Another Space Read-Write   |
|          |            |                  |     | Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycle is a read-write one in which continuous access cycles switch between different spaces. |
|          |            |                  |     | 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock)   |
|          |            |                  |     | 001: 1 idle cycle inserted (setting prohibited when two-<br>times multiplication has been set for the<br>peripheral clock)   |
|          |            |                  |     | 010: 2 idle cycles inserted  |
|          |            |                  |     | 011: 4 idle cycles inserted  |
|          |            |                  |     | 100: 6 idle cycles inserted  |
|          |            |                  |     | 101: 8 idle cycles inserted  |
|          |            |                  |     | 110: 10 idle cycles inserted   |
|          |            |                  |     | 111: 12 idle cycles inserted   |
| 24 to 22 | IWRWS[2:0] | 011              | R/W | Idle Cycles for Read-Write in the Same Space   |
|          |            |                  |     | Specify the number of idle cycles to be inserted after<br>the access to a memory that is connected to the<br>space. The target cycle is a read-write cycle of which<br>continuous access cycles are for the same space.      |
|          |            |                  |     | 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock)   |
|          |            |                  |     | 001: 1 idle cycle inserted (setting prohibited when two-<br>times multiplication has been set for the<br>peripheral clock)   |
|          |            |                  |     | 010: 2 idle cycles inserted  |
|          |            |                  |     | 011: 4 idle cycles inserted  |
|          |            |                  |     | 100: 6 idle cycles inserted  |
|          |            |                  |     | 101: 8 idle cycles inserted  |
|          |            |                  |     | 110: 10 idle cycles inserted   |
|          |            |                  |     | 111: 12 idle cycles inserted   |
|          |            |                  |     |  |

| Bit      | Bit Name   | Initial<br>Value | R/W | Description  |
|----------|------------|------------------|-----|--|
| 21 to 19 | IWRRD[2:0] | 011              | R/W | Idle Cycles for Read-Read in Another Space   |
|          |            |                  |     | Specify the number of idle cycles to be inserted after<br>the access to a memory that is connected to the<br>space. The target cycle is a read-read cycle of which<br>continuous access cycles switch between different<br>spaces. |
|          |            |                  |     | 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock)   |
|          |            |                  |     | 001: 1 idle cycle inserted (setting prohibited when two-<br>times multiplication has been set for the<br>peripheral clock)   |
|          |            |                  |     | 010: 2 idle cycles inserted  |
|          |            |                  |     | 011: 4 idle cycles inserted  |
|          |            |                  |     | 100: 6 idle cycles inserted  |
|          |            |                  |     | 101: 8 idle cycles inserted  |
|          |            |                  |     | 110: 10 idle cycles inserted   |
|          |            |                  |     | 111: 12 idle cycles inserted   |
| 18 to 16 | IWRRS[2:0] | 011              | R/W | Idle Cycles for Read-Read in the Same Space  |
|          |            |                  |     | Specify the number of idle cycles to be inserted after<br>the access to a memory that is connected to the<br>space. The target cycle is a read-read cycle of which<br>continuous access cycles are for the same space.             |
|          |            |                  |     | 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock)   |
|          |            |                  |     | 001: 1 idle cycle inserted (setting prohibited when two-<br>times multiplication has been set for the<br>peripheral clock)   |
|          |            |                  |     | 010: 2 idle cycles inserted  |
|          |            |                  |     | 011: 4 idle cycles inserted  |
|          |            |                  |     | 100: 6 idle cycles inserted  |
|          |            |                  |     | 101: 8 idle cycles inserted  |
|          |            |                  |     | 110: 10 idle cycles inserted   |
| -        |            |                  |     | 111: 12 idle cycles inserted   |

| Bit    | Bit Name  | Initial<br>Value | R/W | Description   |
|--------|-----------|------------------|-----|---|
| 15, 14 | _         | All 0            | R   | Reserved  |
|        |           |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 13, 12 | TYPE[1:0] | 00               | R/W | Specify the type of memory connected to a space.  |
|        |           |                  |     | 00: Normal space (SRAM)   |
|        |           |                  |     | 01: Reserved (setting prohibited)   |
|        |           |                  |     | 10: Reserved (setting prohibited)   |
|        |           |                  |     | 11: SRAM with byte selection  |
| 11     | _         | 0                | R   | Reserved  |
|        |           |                  |     | This bit is always read as 0. The write value should always be 0.   |
| 10, 9  | BSZ[1:0]  | 10/01            | R/W | Data Bus Width Specification  |
|        |           |                  |     | Specify the data bus widths of spaces.  |
|        |           |                  |     | 00: Reserved (setting prohibited)   |
|        |           |                  |     | 01: 8-bit size  |
|        |           |                  |     | 10: 16-bit size   |
|        |           |                  |     | 11: Reserved (setting prohibited)   |
|        |           |                  |     | Note: In on-chip RAM disabled mode, the initial data bus widths set in the CS0 to CS3 are specified through the mode pins settings. The mode setting pin data sampled at a power-on reset is reflected to the BSZ1 and BSZ0 bits. In on-chip ROM disabled mode, writing to the BSZ1 and BSZ0 bits in CS0BCR are ignored, but the bus width set in CS1BCR to CS3BCR can be modified. In on-chip ROM enabled mode, the initial data bus widths set in CS0 to CS3 are specified as 16 bits. In this case, the bus width set in CS0BCR to CS3BCR can be modified. |
| 8 to 0 | _         | All 0            | R   | Reserved  |
|        |           |                  |     | These bits are always read as 0. The write value should always be 0.  |

# 10.4.2 CSn Space Wait Control Register (CSnWCR) (n = 0 to 3)

CSnWCR specifies various wait cycles for memory access. Specify CSnBCR first, and then specify CSnWCR.

CSnWCR is initialized to H'00000500 by a power-on reset.

| Bit:           | 31 | 30 | 29 | 28  | 27    | 26  | 25  | 24    | 23  | 22  | 21 | 20  | 19 | 18  | 17      | 16    |
|----------------|----|----|----|-----|-------|-----|-----|-------|-----|-----|----|-----|----|-----|---------|-------|
|                | -  | -  | -  | -   | -     | -   | -   | -     | -   | -   | -  | BAS | -  |     | WW[2:0] |       |
| Initial value: | 0  | 0  | 0  | 0   | 0     | 0   | 0   | 0     | 0   | 0   | 0  | 0   | 0  | 0   | 0       | 0     |
| R/W:           | R  | R  | R  | R   | R     | R   | R   | R     | R   | R   | R  | R/W | R  | R/W | R/W     | R/W   |
|                |    |    |    |     |       |     |     |       |     |     |    |     |    |     |         |       |
| Bit:           | 15 | 14 | 13 | 12  | 11    | 10  | 9   | 8     | 7   | 6   | 5  | 4   | 3  | 2   | 1       | 0     |
|                | 1  | -  | 1  | SW  | [1:0] |     | WR  | [3:0] |     | WM  | -  | -   | -  | 1   | HW      | [1:0] |
| Initial value: | 0  | 0  | 0  | 0   | 0     | 1   | 0   | 1     | 0   | 0   | 0  | 0   | 0  | 0   | 0       | 0     |
| R/W:           | R  | R  | R  | R/W | R/W   | R/W | R/W | R/W   | R/W | R/W | R  | R   | R  | R   | R/W     | R/W   |

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

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 18 to 16 | WW[2:0]  | 000              | R/W | Number of Write Access Wait Cycles   |
|          |          |                  |     | Specify the number of cycles that are necessary for write access.                                      |
|          |          |                  |     | 000: The same cycles as WR[3:0] setting (number of read access wait cycles)                            |
|          |          |                  |     | 001: No cycle (setting prohibited when two-times multiplication has been set for the peripheral clock) |
|          |          |                  |     | 010: 1 cycle   |
|          |          |                  |     | 011: 2 cycles  |
|          |          |                  |     | 100: 3 cycles  |
|          |          |                  |     | 101: 4 cycles  |
|          |          |                  |     | 110: 5 cycles  |
|          |          |                  |     | 111: 6 cycles  |
| 15 to 13 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.                                   |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 12, 11 | SW[1:0]  | 00               | R/W | When normal SRAM connection or SRAM with byte selection connection is used with BAS = 0:   |
|        |          |                  |     | Number of Delay Cycles from Address, CSn     Assertion to RD, WEn Assertion  |
|        |          |                  |     | Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ assertion. |
|        |          |                  |     | 00: 0.5 cycles   |
|        |          |                  |     | 01: 1.5 cycles   |
|        |          |                  |     | 10: 2.5 cycles   |
|        |          |                  |     | 11: 3.5 cycles   |
|        |          |                  |     | For SRAM with byte selection connection with BAS = 1:  |
|        |          |                  |     | Number of Delay Cycles from Address,   |
|        |          |                  |     | Assertion to RD Assertion  |
|        |          |                  |     | Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ assertion.                             |
|        |          |                  |     | 00: 0.5 cycles   |
|        |          |                  |     | 01: 1.5 cycles   |
|        |          |                  |     | 10: 2.5 cycles   |
|        |          |                  |     | 11: 3.5 cycles   |
|        |          |                  |     | <ul> <li>Number of Delay Cycles from Address, CSn</li> </ul>   |
|        |          |                  |     | Assertion to RD/WR Assertion   |
|        |          |                  |     | Specify the number of delay cycles from address and CSn assertion to RD/WR assertion.  |
|        |          |                  |     | 00: 1 cycle  |
|        |          |                  |     | 01: 2 cycles   |
|        |          |                  |     | 10: 3 cycles   |
|        |          |                  |     | 11: 4 cycles   |

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

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 1, 0 | HW[1:0]  | 00               | R/W | When normal SRAM connection or SRAM with byte selection connection is used with BAS = 0:                                 |
|      |          |                  |     | Number of Delay Cycles from RD, WEn Negation<br>to Address, CSn Negation   |
|      |          |                  |     | Specify the number of delay cycles from RD and WEn negation to address and CSn negation.                                 |
|      |          |                  |     | 00: 0.5 cycles   |
|      |          |                  |     | 01: 1.5 cycles   |
|      |          |                  |     | 10: 2.5 cycles   |
|      |          |                  |     | 11: 3.5 cycles   |
|      |          |                  |     | For SRAM with byte selection connection with BAS = 1:  |
|      |          |                  |     | <ul> <li>Number of Delay Cycles from RD Negation to</li> </ul>   |
|      |          |                  |     | Address, CSn Negation  |
|      |          |                  |     | Specify the number of delay cycles from $\overline{\text{RD}}$ negation to address and $\overline{\text{CSn}}$ negation. |
|      |          |                  |     | 00: 0.5 cycles   |
|      |          |                  |     | 01: 1.5 cycles   |
|      |          |                  |     | 10: 2.5 cycles   |
|      |          |                  |     | 11: 3.5 cycles   |
|      |          |                  |     | <ul> <li>Number of Delay Cycles from RD/WR Negation to<br/>Address, CSn Negation</li> </ul>                              |
|      |          |                  |     | Specify the number of delay cycles from RD/ $\overline{WR}$ negation to address and $\overline{CSn}$ negation.           |
|      |          |                  |     | 00: 1 cycle  |
|      |          |                  |     | 01: 2 cycles   |
|      |          |                  |     | 10: 3 cycles   |
|      |          |                  |     | 11: 4 cycles   |

# 10.5 Operation

### 10.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian, in which the 0 address is the most significant byte (MSB) in the byte data. Two data bus widths (8 bits and 16 bits) are available.

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

Tables 10.5 and 10.6 show the relationship between device data width and access unit.

Table 10.5 16-Bit External Device Access and Data Alignment

|                           | Data          | a Bus         | Strobe Signals |        |
|---------------------------|---------------|---------------|----------------|--------|
| Operation                 | D15 to D8     | D7 to D0      | WE1            | WE0    |
| Byte access at 0          | Data 7 to 0   | _             | Assert         | _      |
| Byte access at 1          | _             | Data 7 to 0   | _              | Assert |
| Byte access at 2          | Data 7 to 0   | _             | Assert         | _      |
| Byte access at 3          | _             | Data 7 to 0   | _              | Assert |
| Word access at 0          | Data 15 to 8  | Data 7 to 0   | Assert         | Assert |
| Word access at 2          | Data 15 to 8  | Data 7 to 0   | Assert         | Assert |
| Longword 1st time at 0    | Data 31 to 24 | Data 23 to 16 | Assert         | Assert |
| access at 0 2nd time at 2 | Data 15 to 8  | Data 7 to 0   | Assert         | Assert |

Table 10.6 8-Bit External Device Access and Data Alignment

|                      |               | Data      | a Bus         | Strobe S | 3ignals |
|----------------------|---------------|-----------|---------------|----------|---------|
| Operation            |               | D15 to D8 | D7 to D0      | WE1      | WE0     |
| Byte access at 0     |               | _         | Data 7 to 0   | _        | Assert  |
| Byte access          | at 1          | _         | Data 7 to 0   | _        | Assert  |
| Byte access          | at 2          | _         | Data 7 to 0   | _        | Assert  |
| Byte access          | at 3          | _         | Data 7 to 0   | _        | Assert  |
| Word                 | 1st time at 0 | _         | Data 15 to 8  | _        | Assert  |
| access at 0          | 2nd time at 1 | _         | Data 7 to 0   | _        | Assert  |
| Word                 | 1st time at 2 |           | Data 15 to 8  | _        | Assert  |
| access at 2          | 2nd time at 3 | _         | Data 7 to 0   | _        | Assert  |
| Longword access at 0 | 1st time at 0 | _         | Data 31 to 24 | _        | Assert  |
|                      | 2nd time at 1 | _         | Data 23 to 16 | _        | Assert  |
|                      | 3rd time at 2 | _         | Data 15 to 8  | _        | Assert  |
|                      | 4th time at 3 | _         | Data 7 to 0   | _        | Assert  |

#### 10.5.2 **External Space Interface**

#### **(1) Basic Timing for One-time Multiplication Peripheral Clock**

For access to an external space, this LSI uses strobe signal output in consideration of the fact that mainly static RAM or ROM will be directly connected. Figure 10.2 shows the basic timings. A nowait normal access is completed in two cycles.

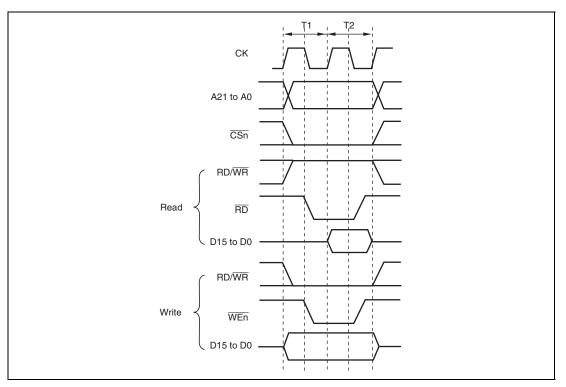


Figure 10.2 Basic Access Timing (Access Wait 0)

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 16 bits are always read in case of a 16-bit apace, and 8 bits in case of an 8-bit space. When writing, only the WEn signal for the byte to be written is asserted.

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

Figures 10.3 and 10.4 show the basic timings of continuous access. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted after the CSn space access to evaluate the external wait (figure 10.3). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 10.4).

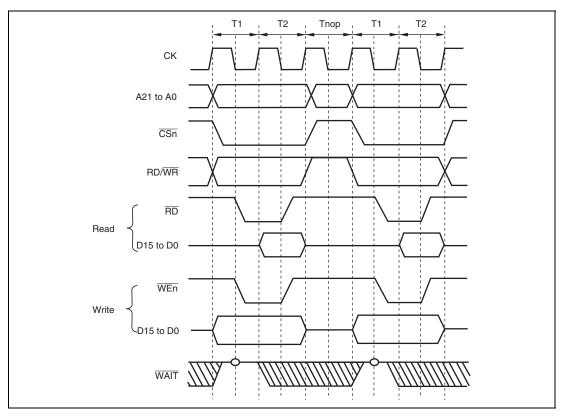


Figure 10.3 Continuous Access 1

Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 0

(Access Wait = 0, Cycle Wait = 0)

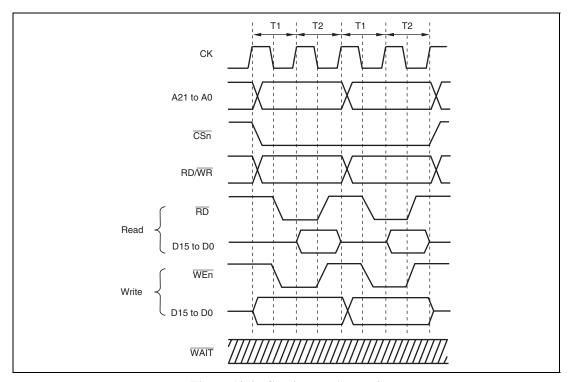


Figure 10.4 Continuous Access 2

Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1

(Access Wait = 0, Cycle Wait = 0)

## (2) Basic Timing for Two-Times Multiplied Peripheral Clock

When two-times multiplication has been set for the peripheral clock, CSnBCR and CSnWCR must be set so that more than one cycle and two cycles are inserted as Tw and Tid, respectively. Figure 10.5 shows the basic timing for two-times multiplied peripheral clock. Figure 10.6 shows a continuous access to normal space for two-times multiplied peripheral clock. Setting the CSnWCR.WR[3:0] bits to 0001 inserts one cycle as Tw (Figure 10.5). Setting the IWW[2:0], IWRWD[2:0], IWRWD[2:0], IWRRD[2:0], and IWRRS[2:0] bits in CSnBCR to 010 inserts two cycles as Tid after access to the CSn space (Figure 10.6). Even when the CSnWCR.WM bit is 0, no Tnop is inserted (see section 10.5.6, Wait between Access Cycles).

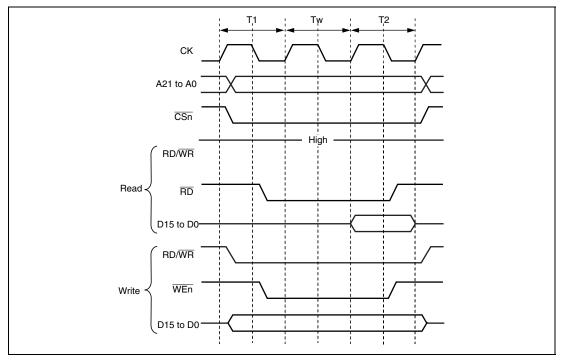


Figure 10.5 Basic Access for Two-Times Multiplied Peripheral Clock (Access Wait 1)

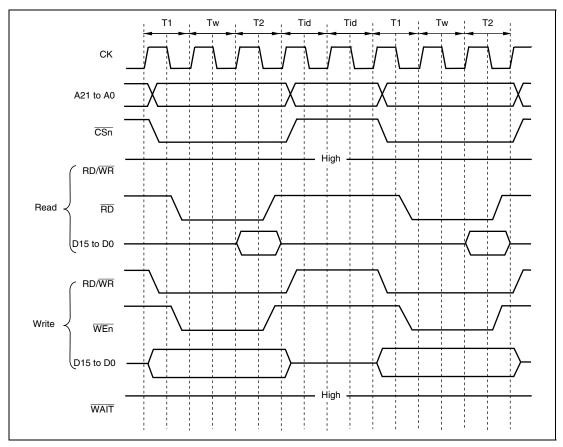


Figure 10.6 Continuous Access for Two-Times Multiplied Peripheral Clock
Bus Width = 16 Bits, Longword Access
(Access Wait = 1, Cycle Wait = 2)

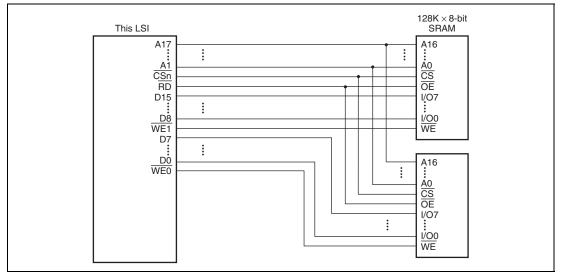


Figure 10.7 Example of 16-Bit Data-Width SRAM Connection

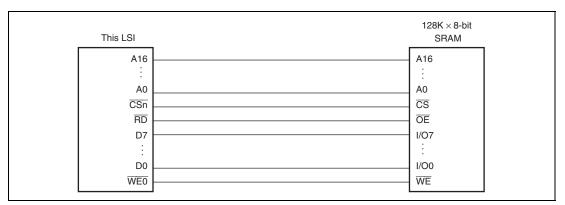


Figure 10.8 Example of 8-Bit Data-Width SRAM Connection

#### 10.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible to insert wait cycles independently in read access and in write access. The specified number of Tw cycles are inserted as wait cycles in access shown in figure 10.9.

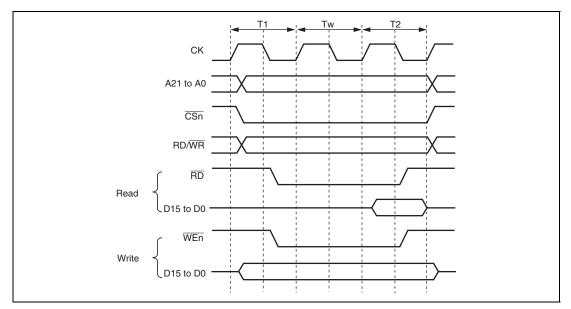


Figure 10.9 Wait Timing for External Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait input WAIT signal is also sampled. WAIT pin sampling is shown in figure 10.10. A 2-cycle wait is specified as a software wait. The WAIT signal is sampled on the falling edge of CK at the transition from the T1 or Tw cycle to the T2 cycle.

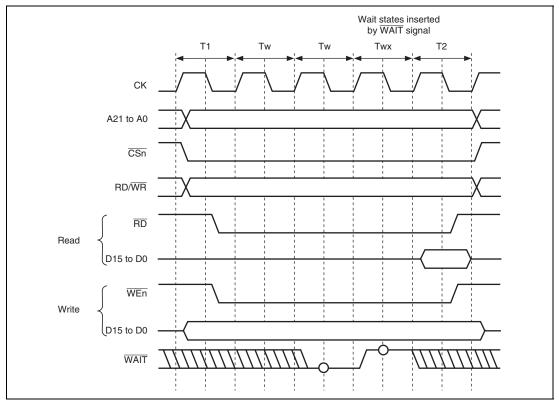
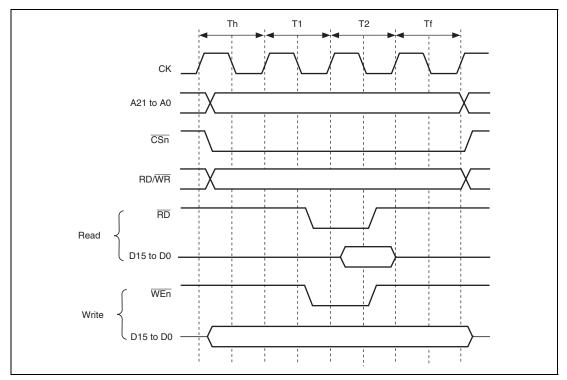


Figure 10.10 Wait Cycle Timing for External Space Access (Wait Cycle Insertion Using  $\overline{WAIT}$  Signal)

#### 

The number of cycles from  $\overline{CSn}$  assertion to  $\overline{RD}$ ,  $\overline{WEn}$  assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from  $\overline{RD}$ ,  $\overline{WEn}$  negation to  $\overline{CSn}$  negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 10.11 shows an example. A Th cycle and a Tf cycle are added before and after an ordinary cycle, respectively. In these cycles,  $\overline{RD}$  and  $\overline{WEn}$  are not asserted, while other signals are asserted. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.



Note: When two-times multiplication has been set for the peripheral clock, Tw must be equal to or more than one cycle and Tid must be equal to or more than two cycles.

### 10.5.5 SRAM Interface with Byte Selection

### (1) Basic Timing for One-Time Multiplied Peripheral Clock

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

When the BAS bit in CSnWCR is cleared to 0 (initial value), the write access timing of the SRAM interface with byte selection is the same as that for the normal space interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the  $\overline{\text{WEn}}$  pin, which is different from that for the normal space interface. The basic access timing is shown in figure 10.12. In write access, data is written to the memory according to the timing of the byte-selection pin ( $\overline{\text{WEn}}$ ). For details, please refer to the Data Sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the  $\overline{\text{WEn}}$  pin and RD/ $\overline{\text{WR}}$  pin timings change. Figure 10.13 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/ $\overline{\text{WR}}$ ). The data hold timing from RD/ $\overline{\text{WR}}$  negation to data write must be acquired by setting the HW1 and HW0 bits in CSnWCR. Figure 10.14 shows the access timing when a software wait is specified.

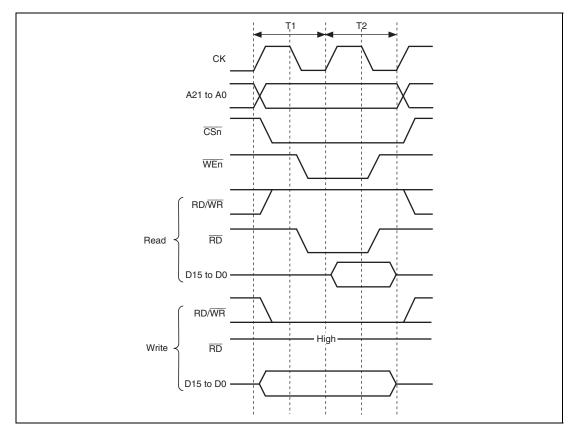


Figure 10.12 Basic Access Timing for SRAM with Byte Selection (BAS = 0)

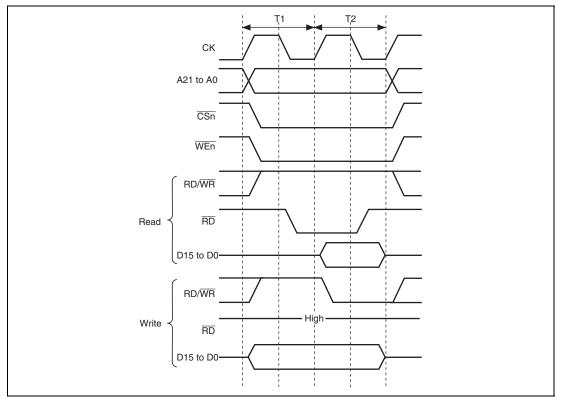


Figure 10.13 Basic Access Timing for SRAM with Byte Selection (BAS = 1)

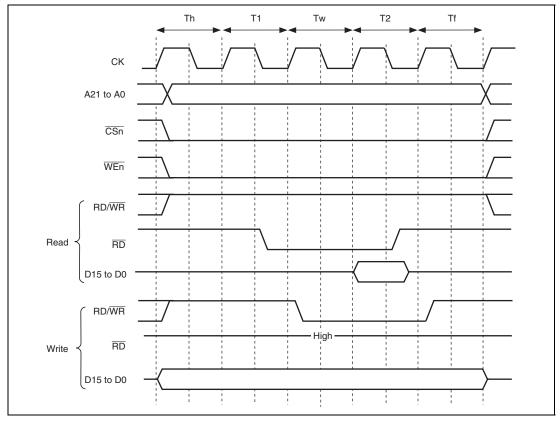


Figure 10.14 Wait Timing for SRAM with Byte Selection (BAS = 1) (SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

# (2) Basic Timing for Two-Times Multiplied Peripheral Clock

When two-times multiplication has been set for the peripheral clock, Tw must be equal to or more than one cycle and Tid must be equal to or more than two cycles as with the normal space interface (for Tid, see figure 10.6). Figure 10.15 shows the basic access timing for the SRAM with byte selection (BAS = 0) on condition that two-times multiplication has been set for the peripheral clock. Figure 10.16 shows the basic access timing in the same situation as figure 10.15 except for BAS = 1.

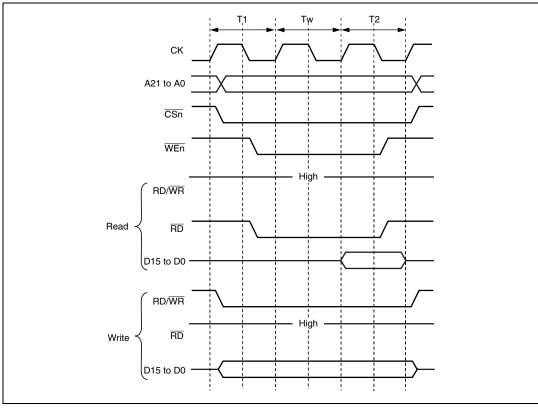


Figure 10.15 Basic Access Timing for SRAM with Byte Selection (BAS = 0) on Condition of Two-Times Multiplied Peripheral Clock

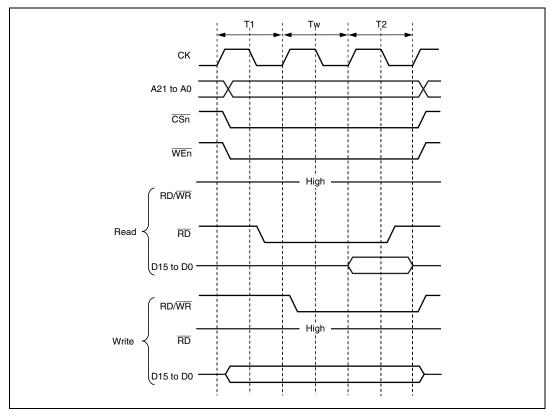


Figure 10.16 Basic Access Timing for SRAM with Byte Selection (BAS = 1) on Condition of **Two-Times Multiplied Peripheral Clock** 

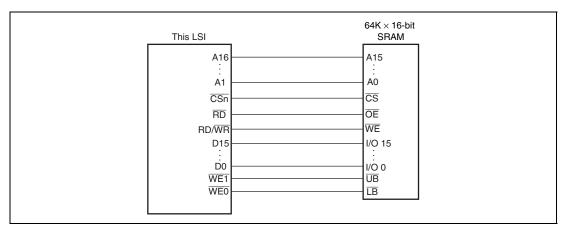


Figure 10.17 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection

### 10.5.6 Wait between Access Cycles

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

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

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

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

Besides the idle cycles between access cycles specified by the above-mentioned registers, idle cycles possibly are inserted to interface with the internal bus. The following gives detailed information about the number of idle cycles from  $\overline{CSn}$  negation to  $\overline{CSn}$  or  $\overline{CSm}$  assertion.

There are six conditions that determine the number of idle cycles on the external bus as shown in table 10.7.

Table 10.7 Conditions for Determining Number of Idle Cycles

| No. | Condition                   | Description  | Range   | Note   |
|-----|-----------------------------|--|---------|--|
| [1] | IW***[2:0] in<br>CSnBCR     | These bits specify the number of idle cycles for external address space access. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS1 space followed by reading other CS space, the bits IWRRD[2:0] in CS1BCR should be set to B'100 to specify six or more idle cycles. This condition generates idle cycles after the access is completed. | 0 to 12 | Do not set 0 for the number of idle cycles between memory types which are not allowed to be accessed successively. |
| [2] | WM in<br>CSnWCR             | This bit enables or disables external $\overline{\text{WAIT}}$ pin. When this bit is set to B'0 (external $\overline{\text{WAIT}}$ enabled), one idle cycle is inserted to check the external $\overline{\text{WAIT}}$ pin input after the access is completed. When this bit is set to B'1 (disabled), no idle cycle is generated.  | 0 or 1  |  |
| [3] | Read data<br>transfer cycle | One idle cycle is inserted after a read access is completed. This idle cycle is not generated for the first or middle cycles in divided access cycles. This is neither generated when the bits HW[1:0] in CSnWCR are set to other value than B'00.   | 0 or 1  |  |

| No. | Condition   | Description  | Range          | Note   |
|-----|---|--|----------------|--|
| [4] | Internal bus idle cycles, etc.                      | External bus access requests from the CPU or DMAC and their results are passed through the internal bus. The external bus enters idle state during internal bus idle cycles or while a bus other than the external bus is being accessed. This condition is not effective for divided access cycles, which are generated by the BSC when the access size is larger than the external data bus width.   | 0 or<br>larger | For CPU read/write → write access cycles, the minimum number of idle cycles is 1.  For CPU read/write → read access cycles, the minimum number of idle cycles is 0.  For DMAC read → write access cycles, the minimum number of idle cycles is 0.  |
| [5] | Write data wait cycles                              | During write access, a write cycle is executed on the external bus only after the write data becomes ready. This write data wait period generates idle cycles before the write cycle. Note that when the previous cycle is a write cycle and the internal bus idle cycles are shorter than the previous write cycle, write data can be prepared in parallel with the previous write cycle and therefore, no idle cycle is generated (write buffer effect). |                | For write → write or write → read access cycles, successive access cycles without idle cycles are frequently available due to the write buffer effect described in the left column. If successive access cycles without idle cycles are not allowed, specify the minimum number of idle cycles between access cycles through CSnBCR. |
| [6] | Idle cycles<br>between<br>different<br>memory types | To ensure the minimum pulse width on the signal-multiplexed pins, idle cycles may be inserted before access after memory types are switched.   | 0 or 1         | One idle cycle is generated<br>between SRAM access or<br>SRAM with byte selection<br>access while BAS = 0 and<br>SRAM with byte selection<br>access while BAS = 1.   |

In the above conditions, a total of four conditions, that is, condition [1], [2], a set of conditions [3] to [5] (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition [6] are generated at the same time. The maximum number of idle cycles among these four conditions become the number of idle cycles on the external bus. To ensure the minimum idle cycles, be sure to make register settings for condition [1].

#### 10.5.7 Others

#### Reset **(1)**

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

#### **(2)** Write Buffer Operation

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

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

The write buffer of the BSC functions in the same way for an access by a bus master other than the CPU such as the DMAC. Accordingly, to perform DMA transfer to external address spaces, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the DMA source and destination addresses exist in external address space, the next write cycle will not be initiated until the previous write cycle is completed.

#### **On-Chip Peripheral Module Access (3)**

To access an on-chip module register, two or more peripheral module clock (P $\phi$ ) cycles are required. Care must be taken in system design.

# 10.6 SRAM Access Timing

### 10.6.1 Standard SRAM

Figure 10.18 shows the SRAM access timing in the CS0 area which corresponds to the basic SRAM access timing shown in figure 10.2, and also shows the timing with wait cycles inserted. In the figure, "SW" denotes a delay cycle specified by the SW1 and SW0 bits in the CSnWCR register, "WW" shows a delay cycle specified by the WW2 to WW0 bits in the CSnWCR register, "HW" indicates a delay cycle specified by the HW1 and HW0 bits in the CSnWCR register, and "WR" denotes a delay cycle specified by the WR3 to WR0 bits in the CSnWCR register.

### 10.6.2 SRAM with Byte Selection (BAS = 0)

Figure 10.19 shows the SRAM access timing in the CS0 area which corresponds to the basic access timing for SRAM with byte selection shown in figure 10.12, and also shows the timing with wait cycles inserted. In the figure, "SW", "WW", "HW", and "WR" indicate the same delay cycles as those specified in section 10.6.1, Standard SRAM.

### 10.6.3 SRAM with Byte Selection (BAS = 1)

Figure 10.20 shows the SRAM access timing in the CS0 area which corresponds to he basic access timing for SRAM with byte selection shown in figure 10.13, and also shows the timing with wait cycles inserted. In the figure, "SW", "WW", "HW", and "WR" indicate the same delay cycles as those specified in section 10.6.1, Standard SRAM.

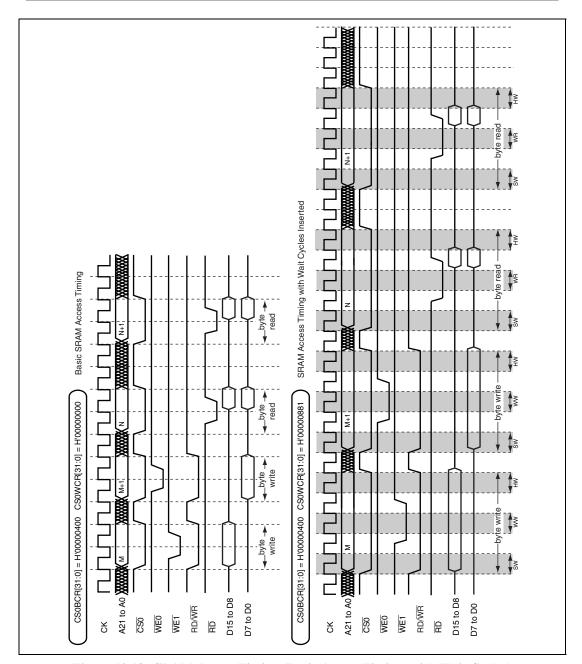


Figure 10.18 SRAM Access Timing (Basic Access Timing, with Wait Cycles)

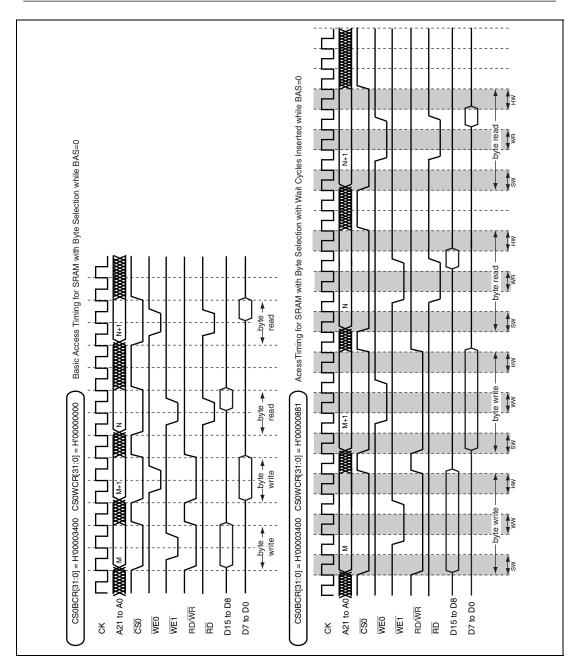


Figure 10.19 Access Timing for SRAM with Byte Selection while BAS=0 (Basic Access Timing, with Wait Cycles)

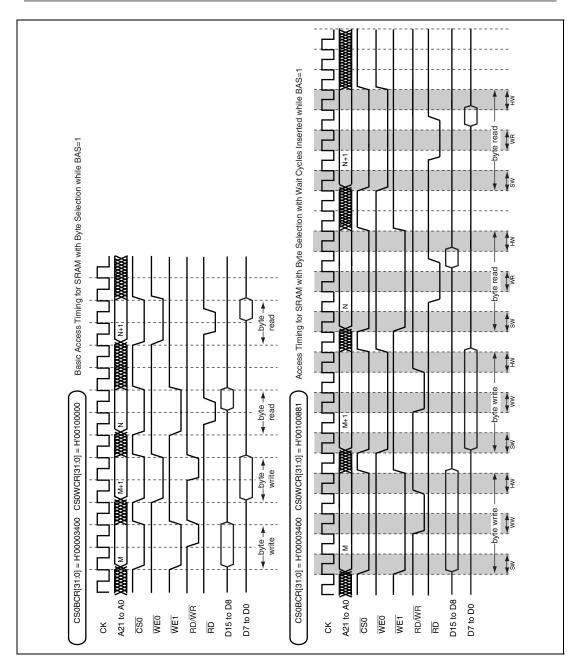


Figure 10.20 Access Timing for SRAM with Byte Selection while BAS=1 (Basic Access Timing, with Wait Cycles)

# Section 11 Direct Memory Access Controller (DMAC)

The DMAC can be used in place of the CPU to perform high-speed transfers between external memory, on-chip memory, external devices that are connected to the external address space, and on-chip peripheral modules.

### 11.1 Features

- Number of channels: Eight channels (channels 0 to 7)
- 4-Gbyte physical address space
- Data transfer unit: Selectable from byte, word (two bytes), longword (four bytes), and 16 bytes (longword × 4)
- Maximum transfer count: 16,777,216 transfers (24 bits)
- Address mode: Dual address mode
- Selectable transfer requests
  - On-chip peripheral module request
  - Auto request

The following modules can issue on-chip peripheral module requests.

- Three RCAN-TL1 sources, ten ADC sources, thirty-five ATU-III sources, and two CMT sources
- Selectable bus modes
  - Cycle stealing mode (normal mode and intermittent mode)
  - Burst mode
- Selectable channel priority levels
  - Two types of fixed modes
  - Round-robin mode
- Interrupt request: An interrupt request can be sent to the CPU on completion of half- or fulldata transfer
- Register reloading functions: The reloading function can be enabled or disabled independently
  for each channel. Using this function, the values in the reload setting registers that can be set
  even during DMA transfer are transferred to the corresponding registers. The following two
  types of reloading functions are supported.
  - Reloading function 1: The DMA source address register, DMA destination address register, DMA transfer count register, and DMA address reload count register are reloaded when the transfer count register reaches 0. The address reload count registers on channels 4 to 7 are also reloaded when the reloading function 2 is enabled.

- Reloading function 2: The DMA source address register and/or DMA destination address register, and DMA address reload count register are reloaded when the address reload count register reaches 0. (Only supported in channels 4 to 7.)
- Continuous transfer when the reloading function 1 is enabled: Selectable between termination and continuation of transfer when the transfer counter reaches 0, on channels the reloading function 1 is enabled for.
- Integer to floating point conversion function: In the integer format to be converted, the data
  stored in 14 bits from the left edge of a 16-bit register is treated as unsigned data, and the
  decimal point is assumed to be located to the left of the MSB. After conversion, the floating
  point format is the single-precision floating-point format based on IEEE754, representing a 32bit value.

Figure 11.1 shows a block diagram of the DMAC.

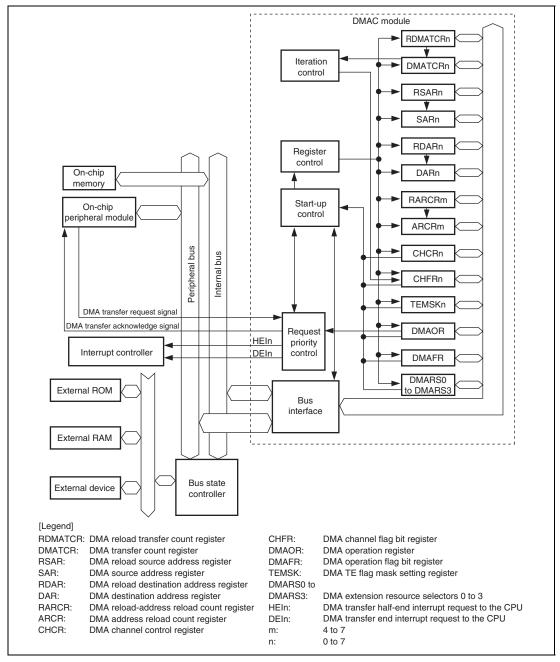


Figure 11.1 Block Diagram of DMAC

# 11.2 Register Descriptions

The DMAC has the registers listed in table 11.1. There are five (channels 0 to 3) or six (channels 4 to 7) control registers, three or four reload registers and one flag bit register for each channel, and one common operation register and one operation flag bit register are used by all channels. In addition, there is one extension resource selector per two channels. Notation for the registers in table 11.1 takes the form XXXN, where XXX indicates the register name and N indicates the channel number. For example, SAR0 denotes SAR for channel 0.

**Table 11.1 Register Configuration** 

| Channel | Register Name                             | Abbreviation | R/W               | Initial Value | Address    | Access<br>Size |
|---------|---|--------------|-------------------|---------------|------------|----------------|
| 0       | DMA source address register 0             | SAR0         | R/W               | H'00000000    | H'FFFE1000 | 16, 32         |
|         | DMA destination address register 0        | DAR0         | R/W               | H'00000000    | H'FFFE1004 | 16, 32         |
|         | DMA transfer count register 0             | DMATCR0      | R/W               | H'00000000    | H'FFFE1008 | 16, 32         |
|         | DMA channel control register 0            | CHCR0        | R/W               | H'00000000    | H'FFFE100C | 8, 16, 32      |
|         | DMA channel flag bit register 0           | CHFR0        | R/W*1             | H'00          | H'FFFE108C | 8              |
|         | DMA TE flag mask setting register 0       | TEMSK0       | R/W* <sup>2</sup> | H'0000        | H'FFFE108E | 8, 16          |
|         | DMA reload source address register 0      | RSAR0        | R/W               | H'00000000    | H'FFFE1100 | 16, 32         |
|         | DMA reload destination address register 0 | RDAR0        | R/W               | H'00000000    | H'FFFE1104 | 16, 32         |
|         | DMA reload transfer count register 0      | RDMATCR0     | R/W               | H'00000000    | H'FFFE1108 | 16, 32         |

| Channel | Register Name                             | Abbreviation | R/W               | Initial Value | Address    | Access<br>Size |
|---------|---|--------------|-------------------|---------------|------------|----------------|
| 1       | DMA source address register 1             | SAR1         | R/W               | H'00000000    | H'FFFE1010 | 16, 32         |
|         | DMA destination address register 1        | DAR1         | R/W               | H'00000000    | H'FFFE1014 | 16, 32         |
|         | DMA transfer count register 1             | DMATCR1      | R/W               | H'00000000    | H'FFFE1018 | 16, 32         |
|         | DMA channel control register 1            | CHCR1        | R/W               | H'00000000    | H'FFFE101C | 8, 16, 32      |
|         | DMA channel flag bit register 1           | CHFR1        | R/W*1             | H'00          | H'FFFE109C | 8              |
|         | DMA TE flag mask setting register 1       | TEMSK1       | R/W* <sup>2</sup> | H'0000        | H'FFFE109E | 8, 16          |
|         | DMA reload source address register 1      | RSAR1        | R/W               | H'00000000    | H'FFFE1110 | 16, 32         |
|         | DMA reload destination address register 1 | RDAR1        | R/W               | H'00000000    | H'FFFE1114 | 16, 32         |
|         | DMA reload transfer count register 1      | RDMATCR1     | R/W               | H'00000000    | H'FFFE1118 | 16, 32         |
| 2       | DMA source address register 2             | SAR2         | R/W               | H'00000000    | H'FFFE1020 | 16, 32         |
|         | DMA destination address register 2        | DAR2         | R/W               | H'00000000    | H'FFFE1024 | 16, 32         |
|         | DMA transfer count register 2             | DMATCR2      | R/W               | H'00000000    | H'FFFE1028 | 16, 32         |
|         | DMA channel control register 2            | CHCR2        | R/W               | H'00000000    | H'FFFE102C | 8, 16, 32      |
|         | DMA channel flag bit register 2           | CHFR2        | R/W*1             | H'00          | H'FFFE10AC | 8              |
|         | DMA TE flag mask setting register 2       | TEMSK2       | R/W* <sup>2</sup> | H'0000        | H'FFFE10AE | 8, 16          |
|         | DMA reload source address register 2      | RSAR2        | R/W               | H'00000000    | H'FFFE1120 | 16, 32         |
|         | DMA reload destination address register 2 | RDAR2        | R/W               | H'00000000    | H'FFFE1124 | 16, 32         |
|         | DMA reload transfer count register 2      | RDMATCR2     | R/W               | H'00000000    | H'FFFE1128 | 16, 32         |

| Channel | Register Name                             | Abbreviation | R/W               | Initial Value | Address    | Access<br>Size |
|---------|---|--------------|-------------------|---------------|------------|----------------|
| 3       | DMA source address register 3             | SAR3         | R/W               | H'00000000    | H'FFFE1030 | 16, 32         |
|         | DMA destination address register 3        | DAR3         | R/W               | H'00000000    | H'FFFE1034 | 16, 32         |
|         | DMA transfer count register 3             | DMATCR3      | R/W               | H'00000000    | H'FFFE1038 | 16, 32         |
|         | DMA channel control register 3            | CHCR3        | R/W               | H'00000000    | H'FFFE103C | 8, 16, 32      |
|         | DMA channel flag bit register 3           | CHFR3        | R/W*1             | H'00          | H'FFFE10BC | 8              |
|         | DMA TE flag mask<br>setting register 3    | TEMSK3       | R/W* <sup>2</sup> | H'0000        | H'FFFE10BE | 8, 16          |
|         | DMA reload source address register 3      | RSAR3        | R/W               | H'00000000    | H'FFFE1130 | 16, 32         |
|         | DMA reload destination address register 3 | RDAR3        | R/W               | H'00000000    | H'FFFE1134 | 16, 32         |
|         | DMA reload transfer count register 3      | RDMATCR3     | R/W               | H'00000000    | H'FFFE1138 | 16, 32         |
| 4       | DMA source address register 4             | SAR4         | R/W               | H'00000000    | H'FFFE1040 | 16, 32         |
|         | DMA destination address register 4        | DAR4         | R/W               | H'00000000    | H'FFFE1044 | 16, 32         |
|         | DMA transfer count register 4             | DMATCR4      | R/W               | H'00000000    | H'FFFE1048 | 16, 32         |
|         | DMA channel control register 4            | CHCR4        | R/W               | H'00000000    | H'FFFE104C | 8, 16, 32      |
|         | DMA channel flag bit register 4           | CHFR4        | R/W*1             | H'00          | H'FFFE10CC | 8              |
|         | DMA TE flag mask<br>setting register 4    | TEMSK4       | R/W* <sup>2</sup> | H'0000        | H'FFFE10CE | 8, 16          |
|         | DMA reload source address register 4      | RSAR4        | R/W               | H'00000000    | H'FFFE1140 | 16, 32         |
|         | DMA reload destination address register 4 | RDAR4        | R/W               | H'00000000    | H'FFFE1144 | 16, 32         |
|         | DMA reload transfer count register 4      | RDMATCR4     | R/W               | H'00000000    | H'FFFE1148 | 16, 32         |

| Channel | Register Name                              | Abbreviation | R/W               | Initial Value | Address    | Access<br>Size |  |
|---------|--|--------------|-------------------|---------------|------------|----------------|--|
| 4       | DMA address reload count register 4        | ARCR4        | R/W               | H'0000        | H'FFFE114C | 16, 32         |  |
|         | DMA reload-address reload count register 4 | RARCR4       | R/W               | H'0000        | H'FFFE114E | 16             |  |
| 5       | DMA source address register 5              | SAR5         | R/W               | H'00000000    | H'FFFE1050 | 16, 32         |  |
|         | DMA destination address register 5         | DAR5         | R/W               | H'00000000    | H'FFFE1054 | 16, 32         |  |
|         | DMA transfer count register 5              | DMATCR5      | R/W               | H'00000000    | H'FFFE1058 | 16, 32         |  |
|         | DMA channel control register 5             | CHCR5        | R/W               | H'00000000    | H'FFFE105C | 8, 16, 32      |  |
|         | DMA channel flag bit register 5            | CHFR5        | R/W*1             | H'00          | H'FFFE10DC | 8              |  |
|         | DMA TE flag mask setting register 5        | TEMSK5       | R/W* <sup>2</sup> | H'0000        | H'FFFE10DE | 8, 16          |  |
|         | DMA reload source address register 5       | RSAR5        | R/W               | H'00000000    | H'FFFE1150 | 16, 32         |  |
|         | DMA reload destination address register 5  | RDAR5        | R/W               | H'00000000    | H'FFFE1154 | 16, 32         |  |
|         | DMA reload transfer count register 5       | RDMATCR5     | R/W               | H'00000000    | H'FFFE1158 | 16, 32         |  |
|         | DMA address reload count register 5        | ARCR5        | R/W               | H'0000        | H'FFFE115C | 16, 32         |  |
|         | DMA reload-address reload count register 5 | RARCR5       | R/W               | H'0000        | H'FFFE115E | 16             |  |
| 6       | DMA source address register 6              | SAR6         | R/W               | H'00000000    | H'FFFE1060 | 16, 32         |  |
|         | DMA destination address register 6         | DAR6         | R/W               | H'00000000    | H'FFFE1064 | 16, 32         |  |
|         | DMA transfer count register 6              | DMATCR6      | R/W               | H'00000000    | H'FFFE1068 | 16, 32         |  |
|         | DMA channel control register 6             | CHCR6        | R/W               | H'00000000    | H'FFFE106C | 8, 16, 32      |  |
|         | DMA channel flag bit register 6            | CHFR6        | R/W*1             | H'00          | H'FFFE10EC | 8              |  |

| Channel | Register Name                              | Abbreviation | R/W               | Initial Value | Address    | Access<br>Size |
|---------|--|--------------|-------------------|---------------|------------|----------------|
| 6       | DMA TE flag mask setting register 6        | TEMSK6       | R/W* <sup>2</sup> | H'0000        | H'FFFE10EE | 8, 16          |
|         | DMA reload source address register 6       | RSAR6        | R/W               | H'00000000    | H'FFFE1160 | 16, 32         |
|         | DMA reload destination address register 6  | RDAR6        | R/W               | H'00000000    | H'FFFE1164 | 16, 32         |
|         | DMA reload transfer count register 6       | RDMATCR6     | R/W               | H'00000000    | H'FFFE1168 | 16, 32         |
|         | DMA address reload count register 6        | ARCR6        | R/W               | H'0000        | H'FFFE116C | 16, 32         |
|         | DMA reload-address reload count register 6 | RARCR6       | R/W               | H'0000        | H'FFFE116E | 16             |
| 7       | DMA source address register 7              | SAR7         | R/W               | H'00000000    | H'FFFE1070 | 16, 32         |
|         | DMA destination address register 7         | DAR7         | R/W               | H'00000000    | H'FFFE1074 | 16, 32         |
|         | DMA transfer count register 7              | DMATCR7      | R/W               | H'00000000    | H'FFFE1078 | 16, 32         |
|         | DMA channel control register 7             | CHCR7        | R/W               | H'00000000    | H'FFFE107C | 8, 16, 32      |
|         | DMA channel flag bit register 7            | CHFR7        | R/W*1             | H'00          | H'FFFE10FC | 8              |
|         | DMA TE flag mask setting register 7        | TEMSK7       | R/W* <sup>2</sup> | H'0000        | H'FFFE10FE | 8, 16          |
|         | DMA reload source address register 7       | RSAR7        | R/W               | H'00000000    | H'FFFE1170 | 16, 32         |
|         | DMA reload destination address register_7  | RDAR7        | R/W               | H'00000000    | H'FFFE1174 | 16, 32         |
|         | DMA reload transfer count register 7       | RDMATCR7     | R/W               | H'00000000    | H'FFFE1178 | 16, 32         |
|         | DMA address reload count register 7        | ARCR7        | R/W               | H'0000        | H'FFFE117C | 16, 32         |
|         | DMA reload-address reload count register 7 | RARCR7       | R/W               | H'0000        | H'FFFE117E | 16             |

| Channel | Register Name                     | Abbreviation | R/W               | Initial Value | Address    | Access<br>Size |
|---------|-----------------------------------|--------------|-------------------|---------------|------------|----------------|
| Common  | DMA operation register            | DMAOR        | R/W               | H'0000        | H'FFFE1200 | 8, 16          |
|         | DMA operation flag bit register   | DMAFR        | R/W* <sup>3</sup> | H'00          | H'FFFE1204 | 8              |
| 0 and 1 | DMA extension resource selector 0 | DMARS0       | R/W               | H'0000        | H'FFFE1300 | 8, 16          |
| 2 and 3 | DMA extension resource selector 1 | DMARS1       | R/W               | H'0000        | H'FFFE1304 | 8, 16          |
| 4 and 5 | DMA extension resource selector 2 | DMARS2       | R/W               | H'0000        | H'FFFE1308 | 8, 16          |
| 6 and 7 | DMA extension resource selector 3 | DMARS3       | R/W               | H'0000        | H'FFFE130C | 8, 16          |

- Notes: 1. Only writing a 0 to the HE and TE bits in CHFRn is allowed after reading a 1, in order to clear the flags.
  - 2. TEMSKn includes a write key code. Write access is made only in word and allowed only if the upper byte, i.e. key code, matches H'5B.
  - 3. Only writing a 0 to the AE and NMIF bits in DMAFR is allowed after reading a 1, in order to clear the flags.

## 11.2.1 DMA Source Address Registers 0 to 7 (SAR0 to SAR7)

SAR is a 32-bit readable/writable register that specifies the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address.

To transfer data in words (two bytes), in longwords (four bytes), or in 16-byte units, specify the address with 2-byte, 4-byte, or 16-byte address boundary.

SAR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

| Bit:                   | 31       | 30       | 29       | 28       | 27       | 26       | 25       | 24       | 23       | 22       | 21       | 20       | 19       | 18       | 17       | 16       |
|------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | 0<br>R/W |
| Bit:                   | 15       | 14       | 13       | 12       | 11       | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | 0<br>R/W |

### 11.2.2 DMA Destination Address Registers 0 to 7 (DAR0 to DAR7)

DAR is a 32-bit readable/writable register that specifies the destination address of a DMA transfer. During a DMA transfer, these registers indicate the next destination address.

To transfer data in words (two bytes), in longwords (four bytes), or in 16-byte units, specify the address with 2-byte, 4-byte, or 16-byte address boundary.

DAR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

| Bit:                   | 31       | 30       | 29       | 28       | 27       | 26       | 25       | 24       | 23       | 22       | 21       | 20       | 19       | 18       | 17       | 16       |
|------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | 0<br>R/W |
| Bit:                   | 15       | 14       | 13       | 12       | 11       | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | 0<br>R/W |

#### 11.2.3 DMA Transfer Count Registers 0 to 7 (DMATCR0 to DMATCR7)

DMATCR is a 32-bit readable/writable register that specifies the number of DMA transfers. The transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'000000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits of DMATCR are always read as 0, and the write value should always be 0. To transfer data in 16-byte units, one 16-byte transfer (128 bits) counts one.

DMATCR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                | -   | -   | -   | -   | -   | -   | -   | -   |     |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R   | R   | R   | R   | R   | R   | R   | R   | R/W |
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |

#### 11.2.4 DMA Channel Control Registers 0 to 7 (CHCR0 to CHCR7)

CHCR is a 32-bit readable/writable register that controls the DMA transfer mode.

The TC[0], RLD2[1:0], and IFT bits can be read and written to in channels 4 to 7, but they are reserved in channels 0 to 3.

CHCR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

| Bit:           | 31  | 30    | 29  | 28    | 27   | 26    | 25   | 24  | 23 | 22 | 21  | 20  | 19    | 18  | 17 | 16  |
|----------------|-----|-------|-----|-------|------|-------|------|-----|----|----|-----|-----|-------|-----|----|-----|
|                | TC[ | 1:0]  | 1   | RLD1  | RLD2 | [1:0] | -    | IFT | -  | -  | -   | -   | -     | HIE | -  | -   |
| Initial value: | 0   | 0     | 0   | 0     | 0    | 0     | 0    | 0   | 0  | 0  | 0   | 0   | 0     | 0   | 0  | 0   |
| R/W:           | R/W | R/W   | R   | R/W   | R/W  | R/W   | R    | R/W | R  | R  | R   | R   | R     | R/W | R  | R   |
| Bit:           | 15  | 14    | 13  | 12    | 11   | 10    | 9    | 8   | 7  | 6  | 5   | 4   | 3     | 2   | 1  | 0   |
|                | DM  | [1:0] | SM  | [1:0] |      | RS    | 3:0] |     | -  | -  | ТВ  | TS  | [1:0] | ΙE  | -  | DE  |
| Initial value: | 0   | 0     | 0   | 0     | 0    | 0     | 0    | 0   | 0  | 0  | 0   | 0   | 0     | 0   | 0  | 0   |
| R/W:           | R/W | R/W   | R/W | R/W   | R/W  | R/W   | R/W  | R/W | R  | R  | R/W | R/W | R/W   | R/W | R  | R/W |

| Bit    | Bit Name | Initial<br>Value | R/W | Descriptions   |
|--------|----------|------------------|-----|--|
| 31, 30 | TC[1:0]  | 00               | R/W | Transfer Count Mode  |
|        |          |                  |     | Specifies whether to transmit data once, for the count specified in DMATCR, or for the count specified in ARCR by one transfer request.  |
|        |          |                  |     | The TC[0] bit is valid only in CHCR4 to CHCR7 and reserved in CHCR0 to CHCR3. The bit in CHCR0 to CHCR3 is always read as 0 and the write value should always be 0.  |
|        |          |                  |     | When these bits are set to a value other than B'01, the TB bit must not be set to 1 (burst mode).  |
|        |          |                  |     | When these bits are set to B'01, RLD2[1] and RLD2[0] bits should be a value other than B'00 to enable the reloading function 2. When these bits are set to B'00, operation is not guaranteed.  |
|        |          |                  |     | 00: Transmits data once by one transfer request  |
|        |          |                  |     | 01: Transmits data for the count specified in ARCR by one transfer request   |
|        |          |                  |     | <ol> <li>Transmits data for the count specified in DMATCR<br/>by one transfer request</li> </ol>   |
|        |          |                  |     | 11: Setting prohibited   |
| 29     | _        | 0                | R   | Reserved   |
|        |          |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 28     | RLD1     | 0                | R/W | Reloading Function 1 Enable or Disable   |
|        |          |                  |     | Specifies whether the reloading function 1 is to be enabled or disabled. When this function is enabled, SAR, DAR, DMATCR or ARCR is reloaded when DMATCR is changed to 0. Note that ARCR is reloaded on channels 4-7 only when the reloading function 2 is also enabled. |
|        |          |                  |     | 0: Disables the reloading function 1   |
|        |          |                  |     | 1: Enables the reloading functioning 1   |

| Bit    | Bit Name  | Initial<br>Value | R/W | Descriptions   |
|--------|-----------|------------------|-----|--|
| 27, 26 | RLD2[1:0] | 00               | R/W | Reloading Function 2 Enable or Disable   |
|        |           |                  |     | Specifies whether the reloading function 2 is to be enabled or disabled. When this function is enabled, SAR, DAR, or ARCR is reloaded when ARCR is changed to 0. |
|        |           |                  |     | These bits are valid only in CHCR4 to CHCR7; in CHCR0 to CHCR3, these bits are reserved. These bits are always read as 0 and the write value should always be 0. |
|        |           |                  |     | 00: Disables reloading function 2  |
|        |           |                  |     | 01: Enables reloading function 2. DAR and ARCR are reloaded.   |
|        |           |                  |     | <ol> <li>Enables reloading function 2. SAR and ARCR are<br/>reloaded.</li> </ol>   |
|        |           |                  |     | 11: Enables the reloading function 2. SAR, DAR, and<br>ARCR are reloaded.  |
| 25     | _         | 0                | R   | Reserved   |
|        |           |                  |     | This bit is always read as 0. The write value should always be 0.  |

| Bit      | Bit Name | Initial<br>Value | R/W | Descriptions   |
|----------|----------|------------------|-----|--|
| 24       | IFT      | 0                | R/W | Integer to Floating-Point Conversion Function Enable   |
|          |          |                  |     | Specifies whether the function converting an integer to the floating-point format is enabled or disabled.  |
|          |          |                  |     | The integer format in which the 14-bit data left-aligned in a 16-bit register is treated as unsigned data and the decimal point is assumed to be located to the left of the MSB can be treated.  |
|          |          |                  |     | After conversion, the integer is represented by a 32-bit single-precision floating-point format based on IEEE 754 (the exponent has a bias of H'7F and the mantissa is represented by a form of 1.xxx, where the first digit 1 is an implicit leading bit).  |
|          |          |                  |     | This bit is valid only in CHCR4 to CHCR7; in CHCR0 to CHCR3, this bit is reserved. This bit is always read as 0 and the write value should always be 0.  |
|          |          |                  |     | If the integer to floating-point conversion function is enabled, the unit of data transfer should be specified as a word (TS in CHCR = B'01). In such a case, a 16-bit value is read as the source of transfer, an integer to floating-point conversion is performed, and the resulting 32-bit value is written. Consequently, the source register is updated by word access, and the destination address is updated by longword access. In addition, if the destination address is not a longword-aligned address, an address error may result. |
|          |          |                  |     | Notice that when the integer to floating-point conversion is enabled, the conversion requires cycles. Therefore, the internal operation within the DMAC will be read (conversion source to inside the DMAC) $\rightarrow$ write (inside the DMAC to conversion source), if a conversion is not required, and read $\rightarrow$ Nop (conversion processing) $\rightarrow$ write, if a conversion is required.  |
|          |          |                  |     | Disables the integer to floating-point conversion function   |
|          |          |                  |     | Enables the integer to floating-point conversion function  |
| 23 to 19 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.   |

| Bit    | Bit Name | Initial<br>Value | R/W | Descriptions   |
|--------|----------|------------------|-----|--|
| 18     | HIE      | 0                | R/W | Half-End Interrupt Enable  |
|        |          |                  |     | Specifies whether to issue an interrupt request to the CPU when the transfer count reaches half of the DMATCR value that was specified before transfer starts.                                 |
|        |          |                  |     | When the HE bit in CHFR is set to 1, the DMAC requests an interrupt to the CPU when the HE bit becomes 1.  |
|        |          |                  |     | 0: Disables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2  |
|        |          |                  |     | 1: Enables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2   |
| 17, 16 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 15, 14 | DM[1:0]  | 00               | R/W | Destination Address Mode   |
|        |          |                  |     | These bits select whether the DMA destination  |
|        |          |                  |     | address is incremented, decremented, or left fixed.  00: Fixed destination address (Setting prohibited in  16-byte transfer)   |
|        |          |                  |     | 01: Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer)   |
|        |          |                  |     | 10: Destination address is decremented (-1 in 8-bit<br>transfer, -2 in 16-bit transfer, -4 in 32-bit transfer,<br>setting prohibited in 16-byte transfer)                                      |
|        |          |                  |     | 11: Setting prohibited   |
| 13, 12 | SM[1:0]  | 00               | R/W | Source Address Mode  |
|        |          |                  |     | These bits select whether the DMA source address is incremented, decremented, or left fixed.   |
|        |          |                  |     | <ol> <li>Fixed source address (Setting prohibited in 16-<br/>byte-unit transfer)</li> </ol>  |
|        |          |                  |     | 01: Source address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte-unit transfer)   |
|        |          |                  |     | <ol> <li>Source address is decremented (–1 in byte-unit<br/>transfer, –2 in word-unit transfer, –4 in longword-<br/>unit transfer, setting prohibited in 16-byte-unit<br/>transfer)</li> </ol> |
|        |          |                  |     | 11: Setting prohibited   |

| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions   |
|---------|----------|------------------|-----|--|
| 11 to 8 | RS[3:0]  | 0000             | R/W | Resource Select  |
| 11.00   | 110[0.0] | 0000             |     | These bits specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state when DMA enable bit (DE) is cleared to 0. |
|         |          |                  |     | 0000: Initial value (when the resource is not selected)  |
|         |          |                  |     | 0001: Setting prohibited   |
|         |          |                  |     | 0010: Setting prohibited   |
|         |          |                  |     | 0011: Setting prohibited   |
|         |          |                  |     | 0100: Auto request   |
|         |          |                  |     | 0101: Setting prohibited   |
|         |          |                  |     | 0110: Setting prohibited   |
|         |          |                  |     | 0111: Setting prohibited   |
|         |          |                  |     | 1000: DMA extension resource selector  |
|         |          |                  |     | 1001: RCAN-TL1 channel A   |
|         |          |                  |     | 1010: RCAN-TL1 channel B   |
|         |          |                  |     | 1011: RCAN-TL1 channel C   |
|         |          |                  |     | 1100: ADC_A (A/D conversion end)   |
|         |          |                  |     | 1101: ADC_B (A/D conversion end) 1110: Setting prohibited  |
|         |          |                  |     | 1111: Setting prohibited   |
| 7, 6    |          | All 0            | R   | Reserved   |
| 7, 0    |          | All U            | 11  | These bits are always read as 0. The write value should always be 0.   |
| 5       | ТВ       | 0                | R/W | Transfer Bus Mode  |
|         |          |                  |     | Specifies the bus mode when DMA transfers data. Note that the burst mode must not be selected when TC is other than B'10.  |
|         |          |                  |     | 0: Cycle stealing mode   |
|         |          |                  |     | 1: Burst mode  |
| 4, 3    | TS[1:0]  | 00               | R/W | Transfer Size  |
|         |          |                  |     | These bits specify the size of data to be transferred.   |
|         |          |                  |     | Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified.                             |
|         |          |                  |     | 00: Byte units   |
|         |          |                  |     | 01: Word (two bytes) units   |
|         |          |                  |     | 10: Longword (four bytes) units  |
|         |          |                  |     | 11: 16-byte units (four longword transfers)  |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 2   | IE       | 0                | R/W | Interrupt Enable  |
|     |          |                  |     | Specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when the TE bit in CHFR is set to 1.  |
|     |          |                  |     | 0: Disables an interrupt request  |
|     |          |                  |     | 1: Enables an interrupt request   |
| 1   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.   |
| 0   | DE       | 0                | R/W | DMA Enable  |
|     |          |                  |     | Enables or disables the DMA transfer. In auto request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this case, all of the bits TE in CHFR, NMIF and AE in DMAFR must be 0. In a peripheral module request, DMA transfer starts if DMA transfer request is generated by peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE in CHFR, NMIF and AE in DMAFR must be 0 as in the case of auto request mode. Clearing the DE bit to 0 can terminate the DMA transfer. |
|     |          |                  |     | 0: DMA transfer disabled  |
|     |          |                  |     | 1: DMA transfer enabled   |

#### DMA Channel Flag Bit Registers 0 to 7 (CHFR0 to CHFR7) 11.2.5

CHFR is an 8-bit readable/writable register that shows the DMA transfer result.

CHFR is initialized to H'00 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

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

Note: \* To clear the flag, write 0 to the bit to be cleared, only when 1 has been read from the bit when these registers have been read. To the HE or TE bit not to be cleared, only 1 can be written, even if 0 is read from the bit. To the reserved bits, only 0 can be written. When the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally. In this case, if 0 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU. Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

|        |          | Initial |        |   |
|--------|----------|---------|--------|---|
| Bit    | Bit Name | Value   | R/W    | Descriptions  |
| 7 to 5 | _        | All 0   | R      | Reserved  |
|        |          |         |        | These bits are always read as 0. The write value should always be 0.  |
| 4      | HE       | 0       | R/(W)* | Half-End Flag   |
|        |          |         |        | This bit is set to 1 when the transfer count reaches half of the DMATCR value specified before transfer starts.   |
|        |          |         |        | When DMA transfer ends because of an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR before the transfer count reaches half of the initial DMATCR value, the HE bit is not set to 1. If DMA transfer ends due to an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR after the HE bit is set to 1, the bit remains set to 1. |
|        |          |         |        | To clear the HE bit, write 0 to it after HE = 1 is read.<br>Note that the HE bit is never set even after it is cleared<br>when the transfer count is equal to or more than the<br>half of the initial DMATCR value.   |
|        |          |         |        | DMATCR > (DMATCR set before transfer starts)/2     during DMA transfer or after DMA transfer is     terminated  |
|        |          |         |        | 1: DMATCR $\leq$ (DMATCR set before transfer starts)/2  |
|        |          |         |        | [Clearing condition]  |
|        |          |         |        | <ul> <li>Writing 0 after reading HE = 1.</li> </ul>   |

| Bit    | Bit Name | Initial<br>Value | R/W    | Descriptions  |
|--------|----------|------------------|--------|---|
| 3 to 1 | _        | All 0            | R      | Reserved  |
|        |          |                  |        | These bits are always read as 0. The write value should always be 0.  |
| 0      | TE       | 0                | R/(W)* | Transfer End Flag   |
|        |          |                  |        | This bit is set to 1 when DMATCR becomes 0 and DMA transfer ends.   |
|        |          |                  |        | The TE bit is not set to 1 in the following cases.  |
|        |          |                  |        | <ul> <li>When DMA transfer ends due to an NMI interrupt<br/>or DMA address error before DMATCR becomes 0</li> </ul>     |
|        |          |                  |        | <ul> <li>When DMA transfer is ended by clearing the DE bit<br/>and DME bit in DMA operation register (DMAOR)</li> </ul> |
|        |          |                  |        | To clear the TE bit, write 0 after reading TE = 1.  |
|        |          |                  |        | Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled.                                     |
|        |          |                  |        | 0: During the DMA transfer or DMA transfer has been terminated  |
|        |          |                  |        | 1: DMA transfer ends by the specified count (DMATCR = 0)  |
|        |          |                  |        | [Clearing condition]  |
|        |          |                  |        | Writing 0 after reading TE = 1  |

Note: \* Only writing a 0 is allowed after reading a 1, in order to clear the flag.

#### 11.2.6 DMA TE Flag Mask Setting Registers 0 to 7 (TEMSK0 to TEMSK7)

TEMSK is a 16-bit readable/writable register, which contains an 8-bit write key. If the reloading function 1 is enabled, the TE flag setting controls whether to terminate or continue DMA transfer. If the reloading function 1 is disabled, the resister setting is ignored and DMA transfer is terminated when the TE flag is set.

The register is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode but retains its value in module standby mode.

Since TEMSK includes a write key, a write to this register must be performed in word. To rewrite the TEMASK bit value, write a value of H'5B to the TEMKEY in advance. A write in word when a value other than H'5B is written to the TEMKEY bit and a write in byte are all ignored.

A read out from TEMSK can be performed both in word and byte. However, the TEMKEY bits are always read as H'00 since no write is held in these bits.

| Bit:           | 15   | 14   | 13   | 12    | 11      | 10   | 9    | 8    | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0          |
|----------------|------|------|------|-------|---------|------|------|------|---|---|---|---|---|---|---|------------|
|                |      |      |      | TEMKI | EY[7:0] |      |      |      | - | - | - | - | - | - | - | TEM<br>ASK |
| Initial value: | 0    | 0    | 0    | 0     | 0       | 0    | 0    | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0          |
| R/W:           | R/W* | R/W* | R/W* | R/W*  | R/W*    | R/W* | R/W* | R/W* | R | R | R | R | R | R | R | R/W        |

Note: \* Write data is not retained; these bits are always read as 0.

| Bit     | Bit Name | Initial<br>Value | R/W  | Descriptions  |
|---------|----------|------------------|------|---|
| 15 to 8 | TEMKEY   | All 0            | R/W* | TEMSK Write Key Code  |
|         | [7:0]    |                  |      | These bits act as the write key for TEMSK that enables/disables rewriting the TEMASK bit. Since no write value is held, these bits are always read as H'00. |
|         |          |                  |      | H'5B: Enables rewriting the TEMASK bit.   |
|         |          |                  |      | Other than H'5B: Disables rewriting the TEMASK bit.   |
| 7 to 1  | _        | All 0            | R    | Reserved  |
|         |          |                  |      | These bits are always read as 0. The write value should always be 0.  |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions   |
|-----|----------|------------------|-----|--|
| 0   | TEMASK   | 0                | R/W | TE Flag Mask Setting   |
|     |          |                  |     | Selects whether or not to terminate DMA transfer when<br>the TE bit is set to 1. By setting this bit to 1 while the<br>reloading function 1 is enabled, DMA transfer is<br>performed until its transfer request is canceled.   |
|     |          |                  |     | This function can only be enabled when the reloading function 1 is enabled.  |
|     |          |                  |     | When DMA transfer is continued even after the TE flag is set to 1 by setting the TEMASK bit to 1 while reloading function 1 is enabled, the operation of the HE and TE bits of the CHFR register is as follows:  |
|     |          |                  |     | If the HE bit is cleared in the middle of transfer, the bit is again set when the DMATCR value has become equal to or less than the half of its value set before transfer starts.  |
|     |          |                  |     | <ul> <li>If the TE bit is cleared in the middle of transfer, the<br/>bit is again set when the DMATCR value has<br/>become 0 next time.</li> </ul>   |
|     |          |                  |     | <ul> <li>The information of that the set HE or TE bit was<br/>read for clearing is held until the bit is cleared.</li> <li>Therefore, the set HE or TE bit that was read can<br/>be cleared by simply writing a 0 to it even after the<br/>register values are changed by the reloading<br/>function 1.</li> </ul> |
|     |          |                  |     | 0: Terminates DMA transfer when the TE flag is set to 1.   |
|     |          |                  |     | 1: Continues DMA transfer even when the TE flag is set to 1.   |
|     |          |                  |     | Note: When this function is enabled, care must be taken since the bus occupancy of DMA transfer becomes higher. In particular, when auto request is selected as the request type, do not select burst transfer as the bus mode, as correct operation is not guaranteed in this case.                               |

## 11.2.7 DMA Reload Source Address Registers 0 to 7 (RSAR0 to RSAR7)

RSAR is a 32-bit readable/writable register that can be modified even during DMA transfer.

When the reloading function 1 is enabled (RLD1 in CHCR = 1), the RSAR value is written to the source address register (SAR) at the end of the current DMA transfer (DMATCR = 0). When the reloading function 2 is enabled (RLD2[1] bit in CHCR = 1), the RSAR value is written to the source address register (SAR) when the current address reload count register reaches 0 (ARCR = 0). A new value for the next DMA transfer can be preset in RSAR. When both of the reloading functions are disabled, RSAR is ignored.

To transfer data in words (two bytes), in longwords (four bytes), or in 16-byte units, specify the address with 2-byte, 4-byte, or 16-byte address boundary.

RSAR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

| Bit:                   | 31       | 30       | 29       | 28       | 27       | 26       | 25       | 24       | 23       | 22       | 21       | 20       | 19       | 18       | 17       | 16       |
|------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | 0<br>R/W |
| Bit:                   | 15       | 14       | 13       | 12       | 11       | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:         | 0<br>R/W | 0<br>B/W | 0<br>B/W | 0<br>R/W | 0<br>B/W | 0<br>R/W | 0<br>B/W | 0<br>B/W | 0<br>R/W | 0<br>B/W |

#### 11.2.8 DMA Reload Destination Address Registers 0 to 7 (RDAR0 to RDAR7)

RDAR is a 32-bit readable/writable register that can be modified even during DMA transfer.

When the reloading function 1 is enabled (RLD1 in CHCR = 1), the RDAR value is written to the destination address register (DAR) at the end of the current DMA transfer (DMATCR = 0). When the reloading function 2 is enabled (RLD2[0] bit in CHCR = 1), the RDAR value is written to the destination address register (DAR) when the current address reload count register reaches 0 (ARCR = 0). A new value for the next DMA transfer can be preset in RDAR. When both of the reloading functions are disabled, RDAR is ignored.

To transfer data in words (two bytes), in longwords (four bytes), or in 16-byte units, specify the address with 2-byte, 4-byte, or 16-byte address boundary.

RDAR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

| Bit:                   | 31       | 30       | 29       | 28       | 27       | 26       | 25       | 24       | 23       | 22       | 21       | 20       | 19       | 18       | 17       | 16       |
|------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | 0<br>R/W |
| Bit:                   | 15       | 14       | 13       | 12       | 11       | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | 0<br>R/W |

## 11.2.9 DMA Reload Transfer Count Registers 0 to 7 (RDMATCR0 to RDMATCR7)

RDMATCR is a 32-bit readable/writable register that can be modified even during DMA transfer.

When the reloading function 1 is enabled (RLD1 in CHCR = 1), the RDMATCR value is written to the transfer count register (DMATCR) at the end of the current DMA transfer (DMATCR = 0). A new value for the next DMA transfer can be preset in RDMATCR. When the reloading function 1 is disabled (RLD1 in CHCR = 0), RDMATCR is ignored.

The upper eight bits of RDMATCR are always read as 0, and the write value should always be 0.

As in DMATCR, the transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. To transfer data in 16-byte units, one 16-byte transfer (128 bits) counts one.

RDMATCR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                | -   | -   | -   | -   | -   | -   | -   | -   |     |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R   | R   | R   | R   | R   | R   | R   | R   | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |

#### 11.2.10 DMA Address Reload Count Registers 4 to 7 (ARCR4 to ARCR7)

ARCR is a 16-bit readable/writable register.

When the reloading function 2 is enabled (CHCR.RLD2[1:0] = 1), the ARCR register and the source address register (SAR) and/or the destination address register (DAR) are reloaded once ARCR reaches 0. When the reloading function 2 is disabled (CHCR.RLD2[1:0] = 0), this register is ignored. Note that the ARCR register is reloaded only when the reloading function 2 is enabled (CHCR.RLD2[1:0] = 1).

The upper eight bits in ARCR are always read as 0 and the write value should always be 0.

As in DMATCR, the transfer count is 1 when the setting is H'0001, 255 when H'00FF is set, and 256 (the maximum) when H'0000 is set. To transfer data in 16-byte units, one 16-byte transfer (128 bits) counts one.

ARCR is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

When the reloading function 2 is enabled (RLD1 or RLD2[0] bit in CHCR = 1), ARCR and SAR and/or DAR are reloaded when ARCR reaches 0. When the reloading function 2 is disabled (RLD1 and RLD2[0] bits are 0), ARCR is ignored. ARCR is updated only when the reloading function 2 is enabled.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|----------------|----|----|----|----|----|----|---|---|-----|-----|-----|-----|-----|-----|-----|-----|
|                | -  | -  | -  | -  | -  | -  | - | - |     |     |     |     |     |     |     |     |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| DAM.           | D  | D  | D  | D  | D  | D  | D | D | D/M | DAM | DAM | DAM | D/M | DAM | D/M | DAM |

#### 11.2.11 DMA Reload-Address Reload Count Registers 4 to 7 (RARCR4 to RARCR7)

RARCR is a 16-bit readable/writable register that can be modified even during DMA transfer.

When the reloading function 2 is enabled (RLD1 or RLD2[0] bit in CHCR = 1), the RARCR value is written to the DMA address reload count register (ARCR) when the current reload count register reaches 0 (ARCR = 0). A new value for the next DMA transfer can be preset in RARCR. When the reloading function 2 is disabled (RLD1 and RLD2[0] bits in CHCR are 0), RARCR is ignored.

When both reloading functions are enabled, the contents of RARCR is written to the address reload count register (ARCR) on completion of the current DMA transfer (DMATCR = 0).

The upper eight bits of RARCR are always read as 0, and the write value should always be 0.

As in ARCR, the transfer count is 1 when the setting is H'0001, 255 when H'00FF is set, and 256 (the maximum) when H'0000 is set. To transfer data in 16-byte units, one 16-byte transfer (128 bits) counts one.

RARCR is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

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

# 11.2.12 DMA Operation Register (DMAOR)

DMAOR is a 16-bit readable/writable register that specifies the priority level of channels at the DMA transfer.

DMAOR is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

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

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 15, 14 | _        | All 0            | R   | Reserved  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 13, 12 | CMS[1:0] | 00               | R/W | Cycle Stealing Mode Select  |
|        |          |                  |     | These bits select either normal mode or intermittent mode in cycle stealing mode.   |
|        |          |                  |     | It is necessary that the bus modes of all channels be<br>set to cycle stealing mode to make the intermittent<br>mode valid. |
|        |          |                  |     | 00: Normal mode   |
|        |          |                  |     | 01: Setting prohibited  |
|        |          |                  |     | 10: Intermittent mode 16 One DMA transfer for every 16 cycles of P $\phi$ clock.  |
|        |          |                  |     | 11: Intermittent mode 64 One DMA transfer for every 64 cycles of P $\phi$ clock.  |

|        |          | Initial |     |  |
|--------|----------|---------|-----|--|
| Bit    | Bit Name | Value   | R/W | Description  |
| 11, 10 | _        | All 0   | R   | Reserved   |
|        |          |         |     | These bits are always read as 0. The write value should always be 0.   |
| 9, 8   | PR[1:0]  | 00      | R/W | Priority Mode  |
|        |          |         |     | These bits select the priority level between channels when there are transfer requests for multiple channels simultaneously.   |
|        |          |         |     | 00: Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7  |
|        |          |         |     | 01: Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7  |
|        |          |         |     | 10: Setting prohibited   |
|        |          |         |     | 11: Round-robin mode (only supported in CH0 to CH3)  |
| 7 to 1 | _        | All 0   | R   | Reserved   |
|        |          |         |     | These bits are always read as 0. The write value should always be 0.   |
| 0      | DME      | 0       | R/W | DMA Master Enable  |
|        |          |         |     | Enables or disables DMA transfer on all channels. If the DME bit and DE bit in CHCR are set to 1, DMA transfer is enabled. However, transfer is enabled only when the TE bit in CHFR is set to 0 or the RLD1 bit in CHCR and TEMASK bit are all set to 1, of the corresponding channel for transfer. Also, the NMIF and AE bits in DMAFR must be all cleared to 0. |
|        |          |         |     | Clearing the DME bit to 0 discontinues the DMA transfer on all channels.   |
|        |          |         |     | 0: DMA transfer is disabled on all channels  |
|        |          |         |     | 1: DMA transfer is enabled on all channels   |

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

Table 11.2 shows the priority change in each mode (modes 0 to 2) specified by the priority mode bits. In round-robin mode, the channel priority to accept the next transfer request may change in up to three ways according to the transfer end channel.

For example, when the transfer end channel is channel 1, the priority of the channel to accept the next transfer request is specified as CH2 > CH3 > CH0 > CH1 > CH4 > CH5 > CH6 > CH7. When the transfer end channel is any one of the channels 4 to 7, round-robin will not be applied and the priority level is not changed at the end of transfer in the channels 4 to 7.

**Table 11.2 Combinations of Priority Mode Bits** 

|                             | Transfer       | Priorit | y Mode |      | Pr  | iority l | _evel a | t End c | of Trans | sfer |       |
|-----------------------------|----------------|---------|--------|------|-----|----------|---------|---------|----------|------|-------|
|                             | End            | Bits    | ,      | High | •   |          |         |         |          |      | ▶ Low |
| Mode                        | CH No.         | PR1     | PR0    | 0    | 1   | 2        | 3       | 4       | 5        | 6    | 7     |
| Mode 0<br>(fixed<br>mode 1) | Any<br>channel | 0       | 0      | CH0  | CH1 | CH2      | СНЗ     | CH4     | CH5      | CH6  | CH7   |
| Mode 1<br>(fixed<br>mode 2) | Any<br>channel | 0       | 1      | CH0  | CH4 | CH1      | CH5     | CH2     | CH6      | СНЗ  | CH7   |
| Mode 2                      | CH0            | 1       | 1      | CH1  | CH2 | СНЗ      | CH0     | CH4     | CH5      | CH6  | CH7   |
| (round-<br>robin            | CH1            | 1       | 1      | CH2  | СНЗ | CH0      | CH1     | CH4     | CH5      | CH6  | CH7   |
| mode)                       | CH2            | 1       | 1      | CH3  | CH0 | CH1      | CH2     | CH4     | CH5      | CH6  | CH7   |
|                             | CH3            | 1       | 1      | CH0  | CH1 | CH2      | СНЗ     | CH4     | CH5      | CH6  | CH7   |
|                             | CH4            | 1       | 1      | CH0  | CH1 | CH2      | СНЗ     | CH4     | CH5      | CH6  | CH7   |
|                             | CH5            | 1       | 1      | CH0  | CH1 | CH2      | СНЗ     | CH4     | CH5      | CH6  | CH7   |
|                             | CH6            | 1       | 1      | CH0  | CH1 | CH2      | СНЗ     | CH4     | CH5      | CH6  | CH7   |
|                             | CH7            | 1       | 1      | CH0  | CH1 | CH2      | СНЗ     | CH4     | CH5      | CH6  | CH7   |

#### 11.2.13 DMA Operation Flag Bit Register (DMAFR)

DMAFR is an 8-bit readable/writable register that shows the DMA transfer status.

DMAFR is initialized to H'00 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

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

Note: \* To clear the flag, write 0 to the bit to be cleared, only when 1 has been read from the bit when these registers have been read.

To the AE or NMIF bit not to be cleared, only 1 can be written, even if 0 is read from the bit. To the reserved bits, only 0 can be written.

When the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally. In this case, if 0 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU.

Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

| Bit    | Bit Name | Initial<br>Value | R/W    | Description   |
|--------|----------|------------------|--------|---|
| 7 to 5 | _        | All 0            | R      | Reserved  |
|        |          |                  |        | These bits are always read as 0. The write value should always be 0.  |
| 4      | AE       | 0                | R/(W)* | Address Error Flag  |
|        |          |                  |        | Indicates whether an address error has occurred by the DMAC. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading as 1. The DMAC internal operation for an address error is as follows: |
|        |          |                  |        | • No address error: Read (source to DMAC) $\rightarrow$ Write (DMAC to destination)   |
|        |          |                  |        | $\bullet  \text{Address error in source address: Nop} \rightarrow \text{Nop}$   |
|        |          |                  |        | $\bullet  \text{Address error in destination address: Read} \rightarrow \text{Nop}$   |
|        |          |                  |        | 0: No DMAC address error  |
|        |          |                  |        | 1: DMAC address error occurred  |
|        |          |                  |        | [Clearing condition]  |
|        |          |                  |        | • Writing 0 after reading AE = 1  |

| Bit    | Bit Name | Initial<br>Value | R/W    | Description  |
|--------|----------|------------------|--------|--|
| 3 to 1 | _        | All 0            | R      | Reserved   |
|        |          |                  |        | These bits are always read as 0. The write value should always be 0.   |
| 0      | NMIF     | 0                | R/(W)* | NMI Flag   |
|        |          |                  |        | Indicates that an NMI interrupt occurred. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading as 1.   |
|        |          |                  |        | When the NMI is input, the current DMA transfer in progress is continued until one transfer unit of data is completed. When DMAC operation is stopped by the NMI, it can be resumed by clearing the NMIF bit to 0 after reading as 1. Even if the NMI interrupt is input while the DMAC is not in operation, the NMIF bit is set to 1. |
|        |          |                  |        | 0: No NMI interrupt  |
|        |          |                  |        | 1: NMI interrupt occurred  |
|        |          |                  |        | [Clearing condition]   |
|        |          |                  |        | • Writing 0 after reading NMIF = 1   |

Note: \* Only 0 can be written to clear the flag after reading it as 1.

## 11.2.14 DMA Extension Resource Selectors 0 to 3 (DMARS0 to DMARS3)

The DMA extension resource selectors (DMARS) are 16-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 is for channels 0 and 1, DMARS1 is for channels 2 and 3, DMARS2 is for channels 4 and 5, and DMARS3 is for channels 6 and 7. Table 11.3 shows the specifiable combinations.

DMARS can specify transfer requests from 35 ATU-III sources, 8 ADC sources (only for DMARS), and two CMT sources.

DMARS is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

#### DMARS0

| Bit:           | 15  | 14  | 13    | 12       | 11  | 10  | 9     | 8        | 7   | 6   | 5     | 4         | 3   | 2   | 1     | 0        |
|----------------|-----|-----|-------|----------|-----|-----|-------|----------|-----|-----|-------|-----------|-----|-----|-------|----------|
|                |     |     | CH1 M | IID[5:0] |     |     | CH1 R | RID[1:0] |     |     | CH0 N | /IID[5:0] |     |     | CH0 F | RID[1:0] |
| Initial value: | 0   | 0   | 0     | 0        | 0   | 0   | 0     | 0        | 0   | 0   | 0     | 0         | 0   | 0   | 0     | 0        |
| R/W:           | R/W | R/W | R/W   | R/W      | R/W | R/W | R/W   | R/W      | R/W | R/W | R/W   | R/W       | R/W | R/W | R/W   | R/W      |

#### DMARS1

| Bit:           | 15  | 14  | 13    | 12       | 11  | 10  | 9     | 8        | 7   | 6   | 5     | 4         | 3   | 2   | 1     | 0        |
|----------------|-----|-----|-------|----------|-----|-----|-------|----------|-----|-----|-------|-----------|-----|-----|-------|----------|
|                |     |     | CH3 M | IID[5:0] |     |     | CH3 F | IID[1:0] |     |     | CH2 N | /IID[5:0] |     |     | CH2 F | IID[1:0] |
| Initial value: | 0   | 0   | 0     | 0        | 0   | 0   | 0     | 0        | 0   | 0   | 0     | 0         | 0   | 0   | 0     | 0        |
| R/W:           | R/W | R/W | R/W   | R/W      | R/W | R/W | R/W   | R/W      | R/W | R/W | R/W   | R/W       | R/W | R/W | R/W   | R/W      |

#### DMARS2

| Bit:           | 15  | 14  | 13    | 12       | 11  | 10  | 9     | 8        | 7   | 6   | 5     | 4         | 3   | 2   | 1     | 0        |
|----------------|-----|-----|-------|----------|-----|-----|-------|----------|-----|-----|-------|-----------|-----|-----|-------|----------|
|                |     |     | CH5 N | IID[5:0] |     |     | CH5 R | RID[1:0] |     |     | CH4 N | /IID[5:0] |     |     | CH4 F | RID[1:0] |
| Initial value: | 0   | 0   | 0     | 0        | 0   | 0   | 0     | 0        | 0   | 0   | 0     | 0         | 0   | 0   | 0     | 0        |
| R/W:           | R/W | R/W | R/W   | R/W      | R/W | R/W | R/W   | R/W      | R/W | R/W | R/W   | R/W       | R/W | R/W | R/W   | R/W      |

#### DMARS3

| Bit:           | 15  | 14  | 13    | 12       | 11  | 10  | 9     | 8        | 7   | 6   | 5     | 4         | 3   | 2   | 1     | 0        |
|----------------|-----|-----|-------|----------|-----|-----|-------|----------|-----|-----|-------|-----------|-----|-----|-------|----------|
|                |     |     | CH7 M | 1ID[5:0] |     |     | CH7 F | RID[1:0] |     |     | CH6 N | /IID[5:0] |     |     | CH6 F | IID[1:0] |
| Initial value: | 0   | 0   | 0     | 0        | 0   | 0   | 0     | 0        | 0   | 0   | 0     | 0         | 0   | 0   | 0     | 0        |
| R/W:           | R/W | R/W | R/W   | R/W      | R/W | R/W | R/W   | R/W      | R/W | R/W | R/W   | R/W       | R/W | R/W | R/W   | R/W      |

Transfer requests from the various modules specify MID and RID as shown in table 11.3.

**Table 11.3 DMARS Settings** 

| Peripheral Module | Setting Value for One Channel ({MID, RID}) | MID      | RID  | Function |
|-------------------|--|----------|------|----------|
| ATU-III_C0        | H'03                                       | B'000000 | B'11 | _        |
| ATU-III_C1        | H'07                                       | B'000001 | B'11 | _        |
| ATU-III_C2        | H'0B                                       | B'000010 | B'11 | _        |
| ATU-III_D00       | H'13                                       | B'000100 | B'11 | _        |
| ATU-III_D01       | H'17                                       | B'000101 | B'11 | _        |
| ATU-III_D02       | H'1B                                       | B'000110 | B'11 | _        |
| ATU-III_D03       | H'1F                                       | B'000111 | B'11 | _        |
| ATU-III_E0        | H'23                                       | B'001000 | B'11 | _        |
| ATU-III_E1        | H'27                                       | B'001001 | B'11 | _        |
| ATU-III_E2        | H'2B                                       | B'001010 | B'11 | _        |
| ATU-III_E3        | H'2F                                       | B'001011 | B'11 | _        |
| ATU-III_E4        | H'33                                       | B'001100 | B'11 | _        |
| ATU-III_E5        | H'37                                       | B'001101 | B'11 | _        |
| ATU-III_C3        | H'3B                                       | B'001110 | B'11 | _        |
| ATU-III_C4        | H'3F                                       | B'001111 | B'11 | _        |
| ATU-III_G0        | H'43                                       | B'010000 | B'11 | _        |
| ATU-III_G1        | H'47                                       | B'010001 | B'11 | _        |
| ATU-III_G2        | H'4B                                       | B'010010 | B'11 | _        |
| ATU-III_G3        | H'4F                                       | B'010011 | B'11 | _        |
| ATU-III_D10       | H'53                                       | B'010100 | B'11 | _        |
| ATU-III_D11       | H'57                                       | B'010101 | B'11 | _        |
| ATU-III_D12       | H'5B                                       | B'010110 | B'11 | _        |
| ATU-III_D13       | H'5F                                       | B'010111 | B'11 | _        |
| ADC_B AN40        | H'63                                       | B'011000 | B'11 | _        |
| ADC_B AN41        | H'67                                       | B'011001 | B'11 | _        |
| ADC_B AN42        | H'6B                                       | B'011010 | B'11 | _        |
| ADC_B AN43        | H'6F                                       | B'011011 | B'11 |          |
| ADC_B AN44        | H'73                                       | B'011100 | B'11 | _        |

| Peripheral Module | Setting Value for One Channel ({MID, RID}) | MID      | RID  | Function |
|-------------------|--|----------|------|----------|
| ADC_B AN45        | H'77                                       | B'011101 | B'11 |          |
| ADC_B AN46        | H'7B                                       | B'011110 | B'11 |          |
| ADC_B AN47        | H'7F                                       | B'011111 | B'11 |          |
| ATU-III_D20       | H'93                                       | B'100100 | B'11 |          |
| ATU-III_D21       | H'97                                       | B'100101 | B'11 |          |
| ATU-III_D22       | H'9B                                       | B'100110 | B'11 |          |
| ATU-III_D23       | H'9F                                       | B'100111 | B'11 |          |
| ATU-III_D30       | H'D3                                       | B'110100 | B'11 |          |
| ATU-III_D31       | H'D7                                       | B'110101 | B'11 |          |
| ATU-III_D32       | H'DB                                       | B'110110 | B'11 |          |
| ATU-III_D33       | H'DF                                       | B'110111 | B'11 |          |
| ATU-III_J0        | H'E3                                       | B'111000 | B'11 |          |
| ATU-III_J1        | H'E7                                       | B'111001 | B'11 |          |
| ATU-III_G4        | H'EF                                       | B'111011 | B'11 |          |
| ATU-III_G5        | H'F3                                       | B'111100 | B'11 |          |
| CMT_0             | H'FB                                       | B'111110 | B'11 | _        |
| CMT_1             | H'FF                                       | B'111111 | B'11 | _        |

When MID or RID other than the values listed in table 11.3 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits (RS3 to RS0) in CHCR0 to CHCR7 have been set to B'1000. Otherwise, even if DMARS has been set, the transfer request source is not accepted.

# 11.3 Operation

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

#### 11.3.1 Transfer Flow

The DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), and DMA extension resource selector (DMARS) are set for the target transfer conditions.

When the reloading function is used, the DMA address reload count register (ARCR), DMA reload-address reload count register (RARCR), DMA reload source address register (RSAR), DMA reload destination address register (RDAR), and DMA reload transfer count register (RDMATCR) are set as needed. If transfer has been started or an NMI interrupt or an address error has been generated, the DMA channel flag bit register (CHFR) and DMA operation flag bit register (DMAFR) are cleared as needed.

After the above registers are set, the DMAC transfers data according to the following procedure.

- 1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
- 2. When a transfer request comes and transfer is enabled, the DMAC transfers one transfer unit of data (depending on the TS1 and TS0 settings). For an auto request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented by 1 for each transfer. If the RLD2[1:0] bits in CHCR is set to a value other than B'00, the ARCR value will also be decremented by 1. The actual transfer flows vary by address mode and bus mode.
- 3. If the TC[1:0] bits in CHCR are cleared to B'00 and the request is issued from an on-chip peripheral module, the transfer acknowledge signal will be returned to the module.
- 4. If the RLD2[1] and RLD0 bits in CHCR are set to a value other than B'00, the reloading function 2 is activated when transfers have been completed for the count specified by ARCR. When the RLD2[1] bit is set to 1, the reloading operations RSAR → SAR and RARCR → ARCR are performed. When the RLD2[0] bit is set to 1, the reloading operations RDAR → DAR and RARCR → ARCR are performed. If the TC[1:0] bits in CHCR are set to B'01 and the request is issued from an on-chip peripheral module, the transfer acknowledge signal will be returned to the module.

- 5. When half of the specified transfer count is exceeded (when DMATCR reaches half of the initial value), an HEI interrupt is sent to the CPU if the HE bit in CHFR is set to 1 while the HIE bit in CHCR is 1.
- 6. When transfer has been completed for the specified count (when DMATCR reaches 0), the transfer ends normally by setting the TE bit in CHFR to 1. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU. If the RLD1 bit in CHCR is set to 1, the reloading function 1 is activated. The reloading operations RSAR → SAR, RDAR → DAR, RDMATCR → DMATCR, and RARCR → ARCR (only when the reloading function 2 is enabled) are performed. In addition, if RDL1 and TEMASK bits are both set to 1, the operation will be brought back to step 1 and transfer is kept performed until no transfer request is left to be handled. If the TC[0] bit in CHCR is set to 0 and the request is issued from an on-chip peripheral module, the transfer acknowledge signal will be returned to the module.
- 7. When an address error in the DMAC or an NMI interrupt is generated, the transfer is terminated. Transfers are also terminated when the DE bit in CHCR or the DME bit in DMAOR is cleared to 0.

Figure 11.2 is a flowchart of this procedure.

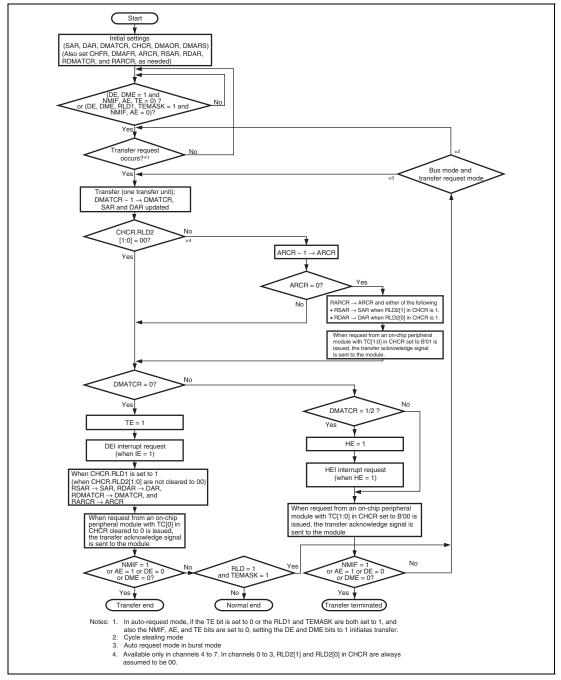


Figure 11.2 DMA Transfer Flowchart

## 11.3.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated in external devices and on-chip peripheral modules that are neither the transfer source nor destination.

Transfers can be requested in two modes: an auto request and an on-chip peripheral module request. The request mode is selected by the RS3 to RS0 bits in CHCR0 to CHCR7 and DMARS0 to DMARS3.

#### (1) Auto-Request Mode

When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits in CHCR0 to CHCR7 and the DME bit in DMAOR are set to 1, the transfer begins so long as the TE bits in CHFR0 to CHFR7, and the AE and NMIF bits in DMAFR are 0.

## (2) On-Chip Peripheral Module Request

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

Signals that request DMA transfer from on-chip peripheral modules include A/D conversion end transfer requests from the ADC, transfer requests from the RCAN-TL1 or ATU-III, and comparematch transfer requests from the CMT.

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

When a transfer is requested from the ADC, the transfer source must be the A/D data register (ADDR). Any address can be specified for data transfer source and destination when a transfer request is generated by the CMT or ATU-III.

Table 11.4 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits

| CHCR    | CHCR DMARS |     | DMA Transfer      |  |                    |                         | _                           |
|---------|------------|-----|-------------------|--|--------------------|-------------------------|-----------------------------|
| RS[3:0] | MID        | RID | Request<br>Source | DMA Transfer<br>Request Signal             | Transfer<br>Source | Transfer<br>Destination | Bus<br>Mode                 |
| 1001    | Any        | Any | RCAN-TL-A         | RMA0 (reception complete)                  | RCAN0<br>(MB0)     | Any                     | Cycle stealing              |
| 1010    | Any        | Any | RCAN-TL-B         | RMB0 (reception complete)                  | RCAN1<br>(MB0)     | Any                     | _                           |
| 1011    | Any        | Any | RCAN-TL-C         | RMC0 (reception complete)                  | RCAN2<br>(MB0)     | Any                     | _                           |
| 1100    | Any        | Any | ADC_A             | ADI0 (scan conversion end)                 | ADR0 to<br>ADR39   | Any                     | Cycle<br>stealing/<br>burst |
| 1101    | Any        | Any | ADC_B             | ADI1 (scan conversion end)                 | ADR40 to<br>ADR47  | Any                     |                             |
| 1000    | 000000     | 11  | ATU-III_C0        | IMIC00 (input<br>capture/compare<br>match) | Any                | Any                     | Cycle<br>stealing/<br>burst |
|         | 000001     | 11  | ATU-III_C1        | IMIC10 (input<br>capture/compare<br>match) | Any                | Any                     |                             |
|         | 000010     | 11  | ATU-III_C2        | IMIC20 (input<br>capture/compare<br>match) | Any                | Any                     | _                           |
|         | 001110     | 11  | ATU-III_C3        | IMIC30 (input<br>capture/compare<br>match) | Any                | Any                     | Cycle<br>stealing/<br>burst |
|         | 001111     | 11  | ATU-III_C4        | IMIC40 (input<br>capture/compare<br>match) | Any                | Any                     |                             |
|         | 000100     | 11  | ATU-III_D00       | UDID00 (down-<br>counter underflow)        | Any                | Any                     |                             |
|         | 000101     | 11  | ATU-III_D01       | UDID01 (down-<br>counter underflow)        | Any                | Any                     |                             |
|         | 000110     | 11  | ATU-III_D02       | UDID02 (down-<br>counter underflow)        | Any                | Any                     | _                           |

Apr 01, 2014

| CHCR          | R DMARS |     | DMA Transfer      |                                     |                    |                         |                             |
|---------------|---------|-----|-------------------|-------------------------------------|--------------------|-------------------------|-----------------------------|
| RS3 to<br>RS0 | MID     | RID | Request<br>Source | DMA Transfer<br>Request Signal      | Transfer<br>Source | Transfer<br>Destination | Bus<br>Mode                 |
| 1000          | 000111  | 11  | ATU-III_D03       | UDID03 (down-<br>counter underflow) | Any                | Any                     | Cycle<br>stealing/          |
|               | 001000  | 11  | ATU-III_E0        | CMIE00 (compare match)              | Any                | Any                     | burst                       |
|               | 001001  | 11  | ATU-III_E1        | CMIE10 (compare match)              | Any                | Any                     | -                           |
|               | 001010  | 11  | ATU-III_E2        | CMIE20 (compare match)              | Any                | Any                     | -                           |
|               | 001011  | 11  | ATU-III_E3        | CMIE30 (compare match)              | Any                | Any                     | -                           |
|               | 001100  | 11  | ATU-III_E4        | CMIE40 (compare match)              | Any                | Any                     | -                           |
|               | 001101  | 11  | ATU-III_E5        | CMIE50 (compare match)              | Any                | Any                     |                             |
|               | 010000  | 11  | ATU-III_G0        | CMIG0 (compare match)               | Any                | Any                     | Cycle<br>stealing/<br>burst |
|               | 010001  | 11  | ATU-III_G1        | CMIG2 (compare match)               | Any                | Any                     |                             |
|               | 010010  | 11  | ATU-III_G2        | CMIG3 (compare match)               | Any                | Any                     |                             |
|               | 010011  | 11  | ATU-III_G3        | CMIG4 (compare match)               | Any                | Any                     |                             |
|               | 010100  | 11  | ATU-III_D10       | UDID10 (down-<br>counter underflow) | Any                | Any                     | Cycle<br>stealing/<br>burst |
|               | 010101  | 11  | ATU-III_D11       | UDID11 (down-<br>counter underflow) | Any                | Any                     |                             |
|               | 010110  | 11  | ATU-III_D12       | UDID12 (down-<br>counter underflow) | Any                | Any                     | -                           |
|               | 010111  | 11  | ATU-III_D13       | UDID13 (down-<br>counter underflow) | Any                | Any                     | -                           |

| CHCR          | CR DMARS |     | DMA Transfer      |                                     |                    |                         |                             |
|---------------|----------|-----|-------------------|-------------------------------------|--------------------|-------------------------|-----------------------------|
| RS3 to<br>RS0 | MID      | RID | Request<br>Source | DMA Transfer<br>Request Signal      | Transfer<br>Source | Transfer<br>Destination | Bus<br>Mode                 |
| 1000          | 011000   | 11  | ADC_B AN40        | ADID40 (interrupt conversion end)   | ADR40              | Any                     | Cycle stealing              |
|               | 011001   | 11  | ADC_B AN41        | ADID41 (interrupt conversion end)   | ADR41              | Any                     | _                           |
|               | 011010   | 11  | ADC_B AN42        | ADID42 (interrupt conversion end)   | ADR42              | Any                     |                             |
|               | 011011   | 11  | ADC_B AN43        | ADID43 (interrupt conversion end)   | ADR43              | Any                     | _                           |
|               | 011100   | 11  | ADC_B AN44        | ADID44 (interrupt conversion end)   | ADR44              | Any                     | <del>-</del><br>-<br>-      |
|               | 011101   | 11  | ADC_B AN45        | ADID45 (interrupt conversion end)   | ADR45              | Any                     |                             |
|               | 011110   | 11  | ADC_B AN46        | ADID46 (interrupt conversion end)   | ADR46              | Any                     |                             |
|               | 011111   | 11  | ADC_B AN47        | ADID47 (interrupt conversion end)   | ADR47              | Any                     |                             |
|               | 100100   | 11  | ATU-III_D20       | UDID20 (down-<br>counter underflow) | Any                | Any                     | Cycle<br>stealing/<br>burst |
|               | 100101   | 11  | ATU-III_D21       | UDID21 (down-<br>counter underflow) | Any                | Any                     |                             |
|               | 100110   | 11  | ATU-III_D22       | UDID22 (down-<br>counter underflow) | Any                | Any                     | _                           |
|               | 100111   | 11  | ATU-III_D23       | UDID23 (down-<br>counter underflow) | Any                | Any                     | _                           |
|               | 110100   | 11  | ATU-III_D30       | UDID30 (down-<br>counter underflow) | Any                | Any                     | Cycle<br>stealing/          |
|               | 110101   | 11  | ATU-III_D31       | UDID31 (down-<br>counter underflow) | Any                | Any                     | - burst                     |

| CHCR          | R DMARS |     | DMA Transfer      |                                     |                    |                         |                             |
|---------------|---------|-----|-------------------|-------------------------------------|--------------------|-------------------------|-----------------------------|
| RS3 to<br>RS0 | MID     | RID | Request<br>Source | DMA Transfer<br>Request Signal      | Transfer<br>Source | Transfer<br>Destination | Bus<br>Mode                 |
| 1000          | 110110  | 11  | ATU-III_D32       | UDID32 (down-<br>counter underflow) | Any                | Any                     | Cycle<br>stealing/<br>burst |
|               | 110111  | 11  | ATU-III_D33       | UDID33 (down-<br>counter underflow) | Any                | Any                     |                             |
|               | 111000  | 11  | ATU-III_J0        | DFIJ0 (FIFO full)                   | Any                | Any                     | _                           |
|               | 111001  | 11  | ATU-III_J1        | DFIJ1 (FIFO full)                   | Any                | Any                     | _                           |
|               | 111011  | 11  | ATU-III_G4        | CMIG4 (compare match)               | Any                | Any                     | _                           |
|               | 111100  | 11  | ATU-III_G5        | CMIG5 (compare match)               | Any                | Any                     | -                           |
|               | 111110  | 11  | CMT_0             | CMI0 (compare match)                | Any                | Any                     | _                           |
|               | 111111  | 11  | CMT_1             | CMI1 (compare match)                | Any                | Any                     |                             |

Note: Set the TC[1:0] bits in CHCR to B'00 only when a DMA transfer request source for which any transfer source and destination can be set is used. Otherwise, operation is not guaranteed. For ADC\_A and ADC\_B, a value other than B'00 can be set.

# 11.3.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. Three modes (fixed mode 1, fixed mode 2, and round-robin mode) are selected using the PR[1:0] bits in DMAOR.

### (1) Fixed Mode

In fixed modes, the priority levels among the channels remain fixed. There are two kinds of fixed modes as follows:

- Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
- Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7

These are selected by the PR[1:0] bits in the DMA operation register (DMAOR).

### (2) Round-Robin Mode

Each time one unit of word, byte, longword, or 16 bytes is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished is rotated to the lowest of the priority order among the four round-robin channels (channels 0 to 4). The priority of the channels other than the round-robin channels (channels 0 to 4) does not change even in round-robin mode. The round-robin mode operation is shown in figure 11.3. The priority in round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 immediately after a reset.

When the round-robin mode has been specified, do not concurrently specify cycle stealing mode and burst mode as the bus modes of any two or more channels.

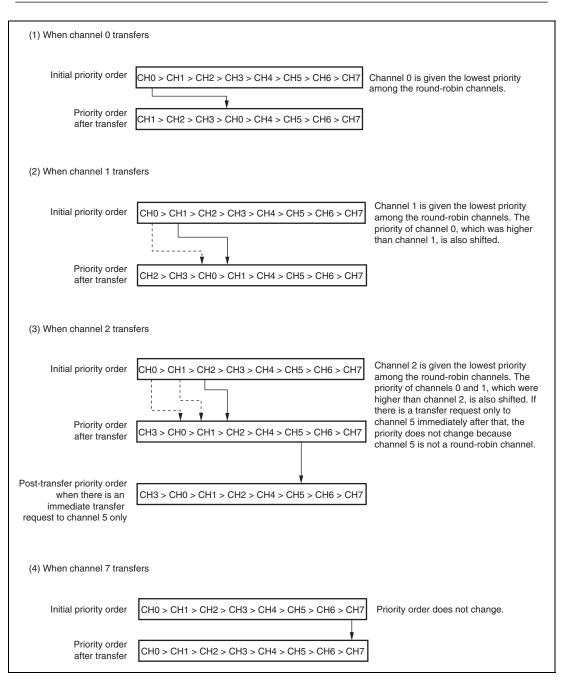


Figure 11.3 Round-Robin Mode

Figure 11.4 shows how the priority order changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

- 1. Transfer requests are generated simultaneously to channels 0 and 3.
- 2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
- 3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
- 4. When the channel 0 transfer ends, channel 0 is given the lowest priority among the round-robin channels.
- 5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
- 6. When the channel 1 transfer ends, channel 1 is given the lowest priority among the round-robin channels.
- 7. The channel 3 transfer begins.
- 8. When the channel 3 transfer ends, channels 3 and 2 are lowered in priority so that channel 3 is given the lowest priority among the round-robin channels.

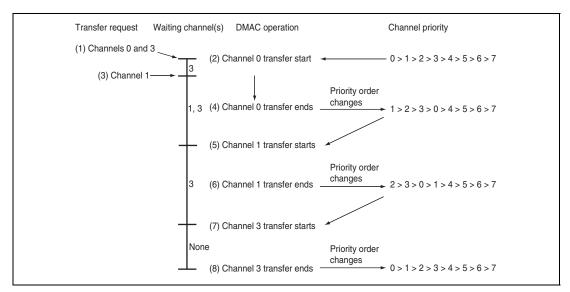


Figure 11.4 Changes in Channel Priority in Round-Robin Mode

# 11.3.4 DMA Transfer Types

This LSI supports DMA transfer in dual address mode. A data transfer timing depends on the bus mode, which is the cycle stealing mode or burst mode. The DMAC supports the transfers shown in table 11.5.

**Table 11.5 Supported DMA Transfers** 

|                           | Transfer Destination |                                  |                                 |                   |  |  |  |
|---------------------------|----------------------|----------------------------------|---------------------------------|-------------------|--|--|--|
| Transfer Source           | External<br>Memory   | External<br>Device* <sup>3</sup> | On-Chip<br>Peripheral<br>Module | On-Chip<br>Memory |  |  |  |
| External memory           | Dual                 | Dual                             | Dual                            | Dual              |  |  |  |
| External device*3         | Dual                 | Dual                             | Dual                            | Dual              |  |  |  |
| On-chip peripheral module | Dual                 | Dual                             | Dual                            | Dual              |  |  |  |
| On-chip memory            | Dual                 | Dual                             | Dual                            | Dual              |  |  |  |

Notes: 1. Dual: Dual address mode

- 16-byte transfer is available only for on-chip peripheral modules that support longword access.
- 3. The external devices are the ones that are connected to the external address space.

### (1) Address Modes

### • Dual Address Mode

In dual address mode, both the transfer source and destination are accessed (selected) by an address. The transfer source and destination can be located externally or internally.

DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC.

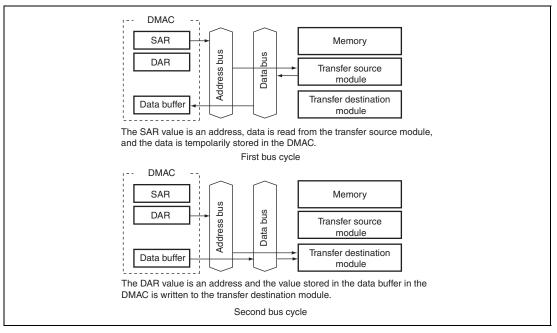


Figure 11.5 Data Flow of Dual Address Mode

Auto request and on-chip peripheral module request are available for the transfer request. Figure 11.6 shows an example of DMA transfer timing in dual address mode.

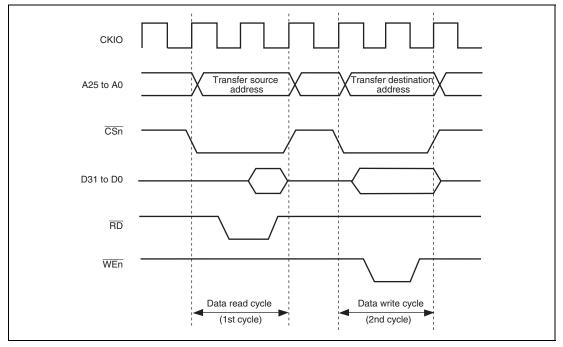


Figure 11.6 Example of DMA Transfer Timing in Dual Mode (Transfer Source: External Memory, Transfer Destination: External Memory)

## (2) Bus Modes

There are two bus modes; cycle stealing and burst. Select the mode by the TB bits in the channel control registers (CHCR).

# • Cycle Stealing Mode

### - Normal mode

In normal mode of cycle stealing, the bus mastership is given to another bus master after a one-transfer-unit (byte, word, longword, or 16-byte unit) DMA transfer. When another transfer request occurs, the bus mastership is obtained from another bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus mastership is passed to another bus master. This is repeated until the transfer end conditions are satisfied.

The cycle-stealing normal mode can be used for any transfer section; transfer request source, transfer source, and transfer destination.

Figure 11.7 shows an example of DMA transfer timing in cycle-stealing normal mode. Dual address mode transfer is performed.

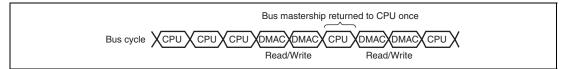


Figure 11.7 DMA Transfer Example in Cycle-Stealing Normal Mode (Dual Address)

Intermittent Mode 16 and Intermittent Mode 64

In intermittent mode of cycle stealing, DMAC returns the bus mastership to other bus master whenever a unit of transfer (byte, word, longword, or 16 bytes) is completed. If the next transfer request occurs after that, DMAC obtains the bus mastership from other bus master after waiting 16 or 64 cycles of Pφ clock. DMAC then transfers data of one unit and returns the bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than the normal mode of cycle stealing.

When DMAC obtains again the bus mastership, DMA transfer may be postponed in case of entry updating due to cache miss.

The cycle-stealing intermittent mode can be used for any transfer section; transfer request source, transfer source, and transfer destination. The bus modes, however, must be cycle stealing mode in all channels.

Figure 11.8 shows an example of DMA transfer timing in cycle-stealing intermittent mode. Dual address mode transfer is performed.

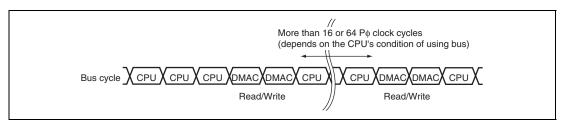


Figure 11.8 Example of DMA Transfer in Cycle-stealing intermittent Mode (Dual Address)

Burst Mode

In burst mode, once the DMAC obtains the bus mastership, it does not release the bus mastership and continues to perform transfer until the transfer end condition is satisfied. Figure 11.9 shows DMA transfer timing in burst mode.



Figure 11.9 DMA Transfer Example in Burst Mode (Dual Address)

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

Table 11.6 shows the relationship between request modes and bus modes by DMA transfer category.

Table 11.6 Relationship of Request Modes and Bus Modes by DMA Transfer Category

| Address<br>Mode | Transfer Category                                       | Request<br>Mode | Bus<br>Mode       | Transfer Size (Bits) | Usable<br>Channels |
|-----------------|---|-----------------|-------------------|----------------------|--------------------|
| Dual            | External memory and external memory                     | All*3           | B/C               | 8/16/32/128          | 0 to 7             |
|                 | External memory and external device*5                   | All*3           | B/C               | 8/16/32/128          | 0 to 7             |
|                 | External device*5 and external device*5                 | All*3           | B/C               | 8/16/32/128          | 0 to 7             |
|                 | External memory and on-chip peripheral module           | All*1           | B/C* <sup>4</sup> | 8/16/32/128*2        | 0 to 7             |
|                 | External device*5 and on-chip peripheral module         | All*1           | B/C* <sup>4</sup> | 8/16/32/128*2        | 0 to 7             |
|                 | On-chip peripheral module and on-chip peripheral module | All*1           | B/C*4             | 8/16/32/128*2        | 0 to 7             |
|                 | On-chip memory and on-chip memory                       | All*3           | B/C               | 8/16/32/128          | 0 to 7             |
|                 | On-chip memory and external device*5                    | All*3           | B/C               | 8/16/32/128          | 0 to 7             |
|                 | On-chip memory and on-chip peripheral module            | All*1           | B/C* <sup>4</sup> | 8/16/32/128*2        | 0 to 7             |
|                 | On-chip memory and external memory                      | All*3           | B/C               | 8/16/32/128          | 0 to 7             |

[Legend]

B: Burst

C: Cycle stealing

Notes: 1. Auto requests and on-chip peripheral module requests are both available. If the request is issued from an on-chip peripheral module, however, the register of the requesting on-chip peripheral module must be the transfer source or destination, other than when the transfer request source is the CMT or ATU-III.

- 2. Access size permitted for the on-chip peripheral module register functioning as the transfer source or transfer destination.
- 3. Auto requests and on-chip peripheral module requests are both available. For on-chip peripheral module requests, however, only the CMT and ATU-III are available as the transfer request source.
- 4. If the request is issued from an on-chip peripheral module, only cycle stealing mode transfer is available other than when the transfer request source is the CMT, ATU-III, 12-bit ADC\_A, or 12-bit ADC\_B.
- 5. The external devices are the ones that are connected to the external address space.

### **Bus Mode and Channel Priority (4)**

In priority fixed mode (CH0 > CH1), when channel 1 is transferring data in burst mode and a request arrives for transfer on channel 0, which has higher-priority, the data transfer on channel 0 will begin immediately. In this case, if the transfer on channel 0 is also in burst mode, the transfer on channel 1 will only resume on completion of the transfer on channel 0.

When channel 0 is in cycle stealing mode, one transfer-unit of data on this channel, which has the higher priority, is transferred. Data is then transferred continuously to channel 1 without releasing the bus. The bus mastership will then switch between the two in this order: channel 0, channel 1, channel 0, channel 1, etc. That is, the CPU cycle after the data transfer in cycle stealing mode is replaced with a burst-mode transfer cycle (priority execution of burst-mode cycle). An example of this is shown in figure 11.10.

When multiple channels are in burst mode, data transfer on the channel that has the highest priority is given precedence. When DMA transfer is being performed on multiple channels, the bus mastership is not released to another bus-master device until all of the competing burst-mode transfers have been completed.

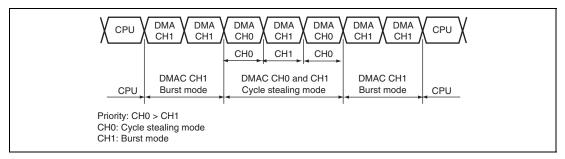


Figure 11.10 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes as shown in figure 11.4. Note that channels in cycle stealing and burst modes must not be mixed.

### 11.4 **Special Operations**

The special operations of the DMAC are described below.

### 11.4.1 Address Error Operation

As explained in the description of the address flag bit (AE) in section 11.2.13, DMA Operation Flag Bit Register (DMAFR), if an address error occurs during transfer operations on the DMAC, the following operation is performed:

- No address error occurred: read (source  $\rightarrow$  DMAC internal)  $\rightarrow$  write (DMAC internal  $\rightarrow$ destination)
- An address error occurred in the source address: Nop  $\rightarrow$  Nop
- An address error occurred in the destination address: read  $\rightarrow$  Nop

### 11.4.2 Operation on NMI

As explained in the description of the NMI flag bit (NMIF) in section 11.2.13, DMA Operation Flag Bit Register (DMAFR), if an NMI occurs during a DMA transfer, the current transfer in progress is continued until one transfer unit of data is completed.

In this case, the DMAC operation can be resumed by clearing the NMIF bit to 0 after reading as 1. The operation can be resumed regardless of whether the reloading functions are enabled or disabled.

### 11.4.3 Operation of Reloading Function 1

Figure 11.11 shows the operation that takes place when the reloading function 1 is enabled and when the reloading function 2 is enabled, as well as the assert timing for the transfer acknowledge signal to an on-chip peripheral module.

Figure 11.12 shows addresses, commands, and data in the internal bus, the states of the peripheral bus, and the update status of the various registers.

Even if the reloading function 1 is enabled, data is not reloaded from RARCR  $\rightarrow$  ARCR. when the reloading function 2 is disabled. The reloading operation is performed only when the reloading functions 1 and 2 are both enabled.

### 11.4.4 Operation of Reloading Function 2

Figure 11.11 shows the operation that takes place when the reloading function 2 is enabled and when the reloading function 1 is enabled, as well as the assert timing for the transfer acknowledge signal to an on-chip peripheral module.

## 11.4.5 Interface with On-Chip Peripheral Module

Figure 11.11 shows the assert timing of the transfer acknowledge signal to an on-chip peripheral module and when the reloading function 1 is enabled and in conjunction with the operation that takes place when the reloading function 2 is enabled.

The assertion timing of the acknowledge signal depends on the settings of the TC[1:0] bits in CHCR.

In channels 0 to 3, the TC[0] bit is always fixed at 0. In channels 4 to 7, setting the TC[1:0] bits to B'11 is prohibited. Further, if the TC[1:0] bits are to be set to B'01, either the RLD2[1] bit or the RLD2[0] bit in CHCR, or both, should be set to 1. If the RLD2[1:0] bits are B'00, the integrity of the resulting operation cannot be guaranteed.

# 11.4.6 Integer to Floating-Point Conversion Operation

As explained in the description of the integer to floating-point function enable bit (IFT) in section 11.2.4, DMA Channel Control Registers 0 to 7 (CHCR0 to CHCR7), when the integer to floating-point conversion function is enabled, the following conversion operations are required:

- No conversion required: read (transfer source to DMAC internal) → write (DMAC internal to transfer destination)
- A conversion required: read  $\rightarrow$  Nop (conversion)  $\rightarrow$  write

The conversion processing involves unusual operations of reading 16-bit data from the transfer source and writing 32-bit data to the transfer destination. For this reason, the TS[1:0] bits in CHCR should be set to B'01 to match the data size at the source of transfer. If the bits are set to any other value, the resulting operation cannot be guaranteed. Further, the transfer destination addresses should be aligned with a 4-byte boundary.

Table 11.7 shows conversion examples on several values. The LSB and the bit next to the LSB of data to be conversion are not converted. The decimal point is indicated by "." in table 11.7.

**Table 11.7 Integer to Floating-Point Conversion Example** 

| Data to be Converted (Hexadecimal) | Converted Data (Hexadecimal)                      |
|------------------------------------|---|
| (0000.) 00 00 0000 0000 0000.      | 0000 0000 0000 0000 0000 0000 00000000 (000000    |
| .0000 0000 0000 01 00 (.0004)      | 0011 1000 1000 0000 0000 0000 00000000 (38800000) |
| .0000 0000 0000 10 00 (.0008)      | 0011 1001 0000 0000 0000 0000 00000000 (39000000) |
| .0100 0000 0000 00 00 (.4000)      | 0011 1110 1000 0000 0000 0000 00000000 (3E800000) |
| .1000 0000 0000 00 (.8000)         | 0011 1111 0000 0000 0000 0000 00000000 (3F000000) |
| .0101 0101 0101 01 00 (.5554)      | 0011 1110 1010 1010 1010 1000 00000000 (3EAAA800) |
| .1010 1010 1010 10 00 (.AAA8)      | 0011 1111 0010 1010 1010 1000 00000000 (3F2AA800) |
| .1100 1100 1100 11 00 (.CCCC)      | 0011 1111 0100 1100 1100 1100 00000000 (3F4CCC00) |
| .0011 0011 0011 00 00 (.3330)      | 0011 1110 0100 1100 1100 0000 00000000 (3F4CC000) |
| .1110 0011 1000 11 00 (.E38C)      | 0011 1111 0110 0011 1000 1100 00000000 (3F638C00) |
| .0001 1100 0111 00 00 (.1C70)      | 0011 1101 1110 0011 1000 0000 00000000 (3DE38000) |
| .1111 0000 1111 00 00 (.F0F0)      | 0011 1111 0111 0000 1111 0000 00000000 (3F70F000) |
| .0000 1111 0000 11 00 (.0F03)      | 0011 1101 0111 0000 0011 0000 00000000 (3D703000) |
| .1111 1111 1111 00 00 (.FFF0)      | 0011 1111 0111 1111 1111 0000 00000000 (3F7FF000) |

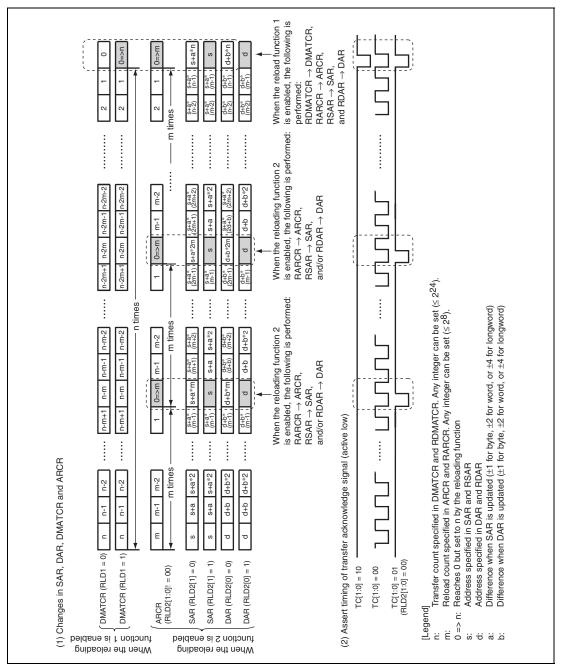


Figure 11.11 Reloading Functions and Timing of Transfer Acknowledge Signal to On-Chip Peripheral Module

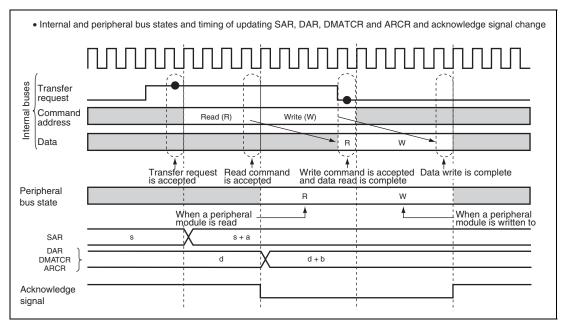


Figure 11.12 Change Timing of Register Contents and Acknowledge Signal

# 11.5 Usage Note

In CHFR0 to CHFR7 and DMAFR, when the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally.

In this case, if 0 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU.

To avoid the malfunction, the clear conditions must be observed, described in the notes of CHFR0 to CHFR7 and DMAFR.

Since the specifications of the flag bits in CHFR0 to CHFR7 and DMAFR differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

# Section 12 Automotive Direct Memory Access Controller (A-DMAC)

The automotive direct memory access controller (A-DMAC) can be used in place of the CPU to perform high-speed data transfers between on-chip peripheral modules and on-chip RAM. The A-DMAC reduces the load of the CPU and improves this LSI operating efficiency.

#### 12.1 **Features**

### Basic functions

Performs high-speed data transfers between on-chip peripheral modules and on-chip RAM in place of the CPU.

Transfer source and destination registers in on-chip peripheral modules are fixed in each channel.

• On-chip peripheral modules

ADC, ATU-III (Timers A, C, F, and G), RSPI, SCI, RCAN-TL1 (hereinafter abbreviated as RCAN)

Alias areas

Areas in on-chip RAM which are used as transfer source and destination areas during data transfer by the A-DMAC are called alias areas.

The start address of an alias area is specified as the start address (H'FFF80000) of the on-chip RAM as default. The start address of the alias area can be changed. The offset from the start address of the alias area for each channel can be specified arbitrarily by address registers for ATU-III (timer G), ADC, RSPI, and SCI channels. The offsets from the start address of the alias area for ATU-III (timers A, C, and F) and RCAN channels are fixed.

- Number of channels: 66
- Transfer requests: fixed for each A-DMAC channel
  - Channel for ADC: ADC (ADC\_A, AN0) interrupt conversion end
  - Channels for ATU-III (Timers A, C, and F): Input capture
  - Channel for ATU-III (Timer G): Compare match
  - Channels for RSPI: Receive buffer full or transmit buffer empty
  - Channels for SCI: Receive data full or transmit data empty
  - Channels for RCAN (reception): Mailbox full
  - Channels for RCAN (transmission): Startup by software

### A-DMAC channel functions

- Channel for ADC: Performs ring-buffer type transfer from AN0 of ADC to on-chip RAM.
- Channel for ATU-III (Timer G): Performs ring-buffer type transfer from on-chip RAM to port G.
- Channels for ATU-III (Timers A, C, and F): Support register reading. Channels for timer A perform input capture register read and transfer the register read value and previous value buffered in the A-DMAC to the alias area. Channels for timers C and F transfer data from input capture register to the alias area. In the channels for timer F, either two register transfer (PWM input waveform measurement mode) or one register transfer (mode other than PWM input waveform measurement mode) can be selected for each channel.
- Channels for RSPI and SCI: Even channels transfer receive data from a register to the alias area. Odd channels transfer transmit data from the alias area to a register.
- Channels for RCAN: A channel for reception transfers data in the mailbox to alias area. A channel for transmission transfers data in the alias area to the mailbox.

### • Specifiable maximum transfer count

- Channels for ADC and ATU-III (Timer G): 1,023; with a reloading function.
- Channels for ATU-III (Timer A, C, and F): Transfer count cannot be specified. If transfer is enabled, data transfer is performed whenever a transfer request is accepted without being restricted by the specified transfer count.
- Channels for RSPI and SCI: 1.023
- Channel for RCAN (reception): Transfer count cannot be specified. Data in the mailbox is transferred whenever a transfer request is accepted without the restriction of the transfer count in the transfer enable state.
- Channel for RCAN (transmission): A transmission of up to 93 mailboxes can be specified simultaneously.
- Interrupt request: Interrupt requests can be masked for each channel.
  - Channels for ATU-III (Timer G) and ADC: Generate no interrupts.
  - Channels for ATU-III (Timers A, C, and F): Can generate an interrupt request to the CPU after a transfer has been completed.
  - Channels for RSPI and SCI: Can generate an interrupt request to the CPU after the specified count of transfers has been completed.
  - Channels for RCAN: A channel for reception does not generate interrupts. A channel for transmission can generate an interrupt request to the CPU after the transmission of the specified mailbox has been completed.
- Address modes
  - Both transfer source and destination addresses are accessed (dual address mode).
- Selectable bus modes

- Cycle steal mode
- Channel priority: The order of channel priority is fixed (channel 0 > channel 1 > ... > channel 74).
- Reloading functions: Supported in channels for ATU-III (timer G) and ADC.

The A-DMAC functions are summarized in table 12.1.

**Table 12.1 A-DMAC Channel Functions** 

|                           | Channels for<br>ATU-III<br>(Timer G) | Channels for ADC                  | Channels for<br>ATU-III<br>(Timers A, C,<br>and F)   | Channels for RSPI  | Channels for SCI   | Channels for RCAN                              |
|---------------------------|--------------------------------------|-----------------------------------|--|--|--|--|
| Channel name              | 0                                    | 1                                 | 2 to 47*1  | 56 to 61   | 62 to 71   | 72, 74* <sup>1</sup>                           |
| Number of channels        | 1                                    | 1                                 | 46   | 6  | 10   | 2  |
| Transfer<br>request       | Timer G2                             | End of ADC_A<br>and AN0           | Ch 2 to 7:<br>Timer A Ch 8<br>to 27: Timer C<br>Ch 28 to 47:<br>Timer F                              | Ch 56 & 57:<br>RSPI_A<br>Ch 58 & 59:<br>RSPI_B,<br>Ch 60 & 61:<br>RSPI_C     | Ch 62 &63:<br>SCI_A<br>Ch 64 & 65:<br>SCI_B<br>Ch 66 to 47:<br>SCI_C<br>Ch 68 & 69:<br>SCI_D<br>Ch 70 & 71:<br>SCI_E | Ch 72: RCAN_A → RCAN_C Ch 74: Software trigger |
| Transfer direction        | RAM → Port                           | Register → RAM                    | Register → RAM   | Even<br>channels:<br>Register →<br>RAM<br>Odd channels:<br>RAM →<br>Register | Even channels: Register → RAM Odd channels: RAM → Register   | Ch 72: Register → RAM Ch 74: RAM → Register    |
| Data transfer length      | 8 bits                               | 16 bits                           | Ch 2 to 7:<br>32 bits × 2<br>Ch 8 to 27:<br>32 bits<br>Ch 28 to 47:<br>32 bits × 2 or<br>32 bits × 1 | 16 bits  | 8 bits   | 20 bytes                                       |
| Maximum<br>transfer count | 1,023 (with a reloading function)    | 1,023 (with a reloading function) | No restriction   | 1,023  | 1,023  | No restriction                                 |

|                      | Channels for<br>ATU-III<br>(Timer G)   | Channels for ADC   | Channels for<br>ATU-III<br>(Timers A, C,<br>and F)                      | Channels for RSPI   | Channels for SCI  | Channels for RCAN   |
|----------------------|--|--|---|---|---|---|
| Interrupt<br>request | No interrupt requests generated  | No interrupt requests generated  | An interrupt<br>request<br>generated at<br>the end of first<br>transfer | An interrupt<br>request<br>generated at<br>the end of<br>transfer<br>(TCR = 0)                    | An interrupt<br>request<br>generated at<br>the end of<br>transfer<br>(TCR = 0)                    | Ch 72: No interrupt requests generated Ch 74: An interrupt request generated at the end of transfer |
| Addressing           | Transfer source<br>address:<br>Incremented,<br>ring-buffer type<br>transfer* <sup>2</sup> ,<br>Transfer<br>destination<br>address: Fixed | Transfer source address: Fixed, Transfer destination address: Incremented, ring-buffer type transfer*2 | Both transfer<br>source and<br>destination<br>addresses:<br>Fixed       | Register<br>address:<br>Fixed,<br>RAM area<br>address:<br>Incremented<br>by two for a<br>transfer | Register<br>address:<br>Fixed,<br>RAM area<br>address:<br>Incremented<br>by two for a<br>transfer | Both transfer<br>source and<br>destination<br>addresses:<br>Incremented                             |

Notes: 1. Channels 48 to 55 and channel 73 are reserved.

2. When TCR = 0, the TCR and alias pointer are reloaded.

Figure 12.1 shows a block diagram of the A-DMAC.

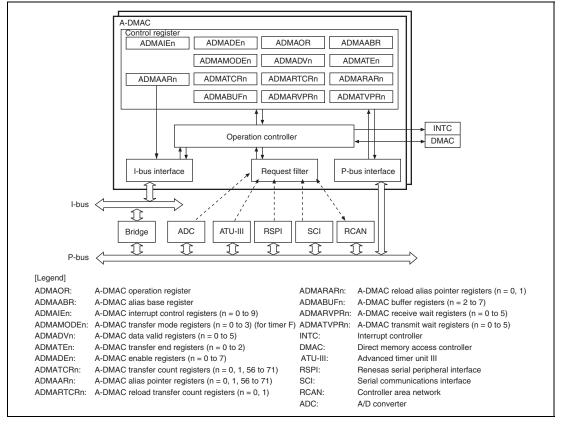


Figure 12.1 Block Diagram of A-DMAC

# 12.1.1 Input/Output Pins

No input/output pins are provided for this module.

# 12.2 Register Descriptions

The A-DMAC has the registers listed in table 12.2.

**Table 12.2 Register Configuration** 

| Channel             | Register Name                       | Abbreviation | R/W     | Initial<br>Value | Address    | Access<br>Size |
|---------------------|-------------------------------------|--------------|---------|------------------|------------|----------------|
| Registers common to | A-DMAC operation register           | ADMAOR       | R/W     | H'00             | H'FFFE6000 | 8              |
| all<br>channels     | A-DMAC alias base register          | ADMAABR      | R/W     | H'00             | H'FFFE6002 | 8              |
| Registers for each  | A-DMAC interrupt control register 0 | ADMAIE0      | R/W     | H'00             | H'FFFE6010 | 8              |
| channel             | A-DMAC interrupt control register 1 | ADMAIE1      | R/W     | H'00             | H'FFFE6011 | 8              |
|                     | A-DMAC interrupt control register 2 | ADMAIE2      | R/W     | H'00             | H'FFFE6012 | 8              |
|                     | A-DMAC interrupt control register 3 | ADMAIE3      | R/W     | H'00             | H'FFFE6013 | 8              |
|                     | A-DMAC interrupt control register 4 | ADMAIE4      | R/W     | H'00             | H'FFFE6014 | 8              |
|                     | A-DMAC interrupt control register 5 | ADMAIE5      | R/W     | H'00             | H'FFFE6015 | 8              |
|                     | A-DMAC interrupt control register 7 | ADMAIE7      | R/W     | H'00             | H'FFFE6017 | 8              |
|                     | A-DMAC interrupt control register 8 | ADMAIE8      | R/W     | H'00             | H'FFFE6018 | 8              |
|                     | A-DMAC interrupt control register 9 | ADMAIE9      | R/W     | H'00             | H'FFFE6019 | 8              |
|                     | A-DMAC data valid register 0        | ADMADV0      | R/(W)*1 | H'00             | H'FFFE6020 | 8              |
|                     | A-DMAC data valid register 1        | ADMADV1      | R/(W)*1 | H'00             | H'FFFE6021 | 8              |
|                     | A-DMAC data valid register 2        | ADMADV2      | R/(W)*1 | H'00             | H'FFFE6022 | 8              |
|                     | A-DMAC data valid register 3        | ADMADV3      | R/(W)*1 | H'00             | H'FFFE6023 | 8              |

| Channel            | Register Name                                 | Abbreviation | R/W     | Initial<br>Value | Address    | Access<br>Size |
|--------------------|---|--------------|---------|------------------|------------|----------------|
| Registers for each | A-DMAC data valid register 4                  | ADMADV4      | R/(W)*1 | H'00             | H'FFFE6024 | 8              |
| channel            | A-DMAC data valid register 5                  | ADMADV5      | R/(W)*1 | H'00             | H'FFFE6025 | 8              |
|                    | A-DMAC transfer end register 0                | ADMATE0      | R/(W)*1 | H'00             | H'FFFE6030 | 8              |
|                    | A-DMAC transfer end register 1                | ADMATE1      | R/(W)*1 | H'00             | H'FFFE6031 | 8              |
|                    | A-DMAC transfer end register 2                | ADMATE2      | R/(W)*1 | H'00             | H'FFFE6032 | 8              |
|                    | A-DMAC enable register 0                      | ADMADE0      | R/W     | H'00             | H'FFFE6040 | 8              |
|                    | A-DMAC enable register 1                      | ADMADE1      | R/W     | H'00             | H'FFFE6041 | 8              |
|                    | A-DMAC enable register 2                      | ADMADE2      | R/W     | H'00             | H'FFFE6042 | 8              |
|                    | A-DMAC enable register 3                      | ADMADE3      | R/W     | H'00             | H'FFFE6043 | 8              |
|                    | A-DMAC enable register 4                      | ADMADE4      | R/W     | H'00             | H'FFFE6044 | 8              |
|                    | A-DMAC enable register 5                      | ADMADE5      | R/W     | H'00             | H'FFFE6045 | 8              |
|                    | A-DMAC enable register 7                      | ADMADE7      | R/W     | H'00             | H'FFFE6047 | 8              |
|                    | A-DMAC transfer mode register 0               | ADMAMODE0    | R/W     | H'00             | H'FFFE6050 | 8              |
|                    | A-DMAC transfer mode register 1               | ADMAMODE1    | R/W     | H'00             | H'FFFE6051 | 8              |
|                    | A-DMAC transfer mode register 2               | ADMAMODE2    | R/W     | H'00             | H'FFFE6052 | 8              |
|                    | A-DMAC transfer count register 0              | ADMATCR0     | R/W     | H'0000           | H'FFFE6060 | 16             |
|                    | A-DMAC reload<br>transfer count<br>register 0 | ADMARTCR0    | R/W     | H'0000           | H'FFFE6062 | 16             |

| Channel            | Register Name                                 | Abbreviation | R/W | Initial<br>Value | Address    | Access<br>Size |
|--------------------|---|--------------|-----|------------------|------------|----------------|
| Registers for each | A-DMAC transfer count register 1              | ADMATCR1     | R/W | H'0000           | H'FFFE6064 | 16             |
| channel            | A-DMAC reload<br>transfer count<br>register 1 | ADMARTCR1    | R/W | H'0000           | H'FFFE6066 | 16             |
|                    | A-DMAC transfer count register 56             | ADMATCR56    | R/W | H'0000           | H'FFFE6070 | 16             |
|                    | A-DMAC transfer count register 57             | ADMATCR57    | R/W | H'0000           | H'FFFE6072 | 16             |
|                    | A-DMAC transfer count register 58             | ADMATCR58    | R/W | H'0000           | H'FFFE6074 | 16             |
|                    | A-DMAC transfer count register 59             | ADMATCR59    | R/W | H'0000           | H'FFFE6076 | 16             |
|                    | A-DMAC transfer count register 60             | ADMATCR60    | R/W | H'0000           | H'FFFE6078 | 16             |
|                    | A-DMAC transfer count register 61             | ADMATCR61    | R/W | H'0000           | H'FFFE607A | 16             |
|                    | A-DMAC transfer count register 62             | ADMATCR62    | R/W | H'0000           | H'FFFE607C | 16             |
|                    | A-DMAC transfer count register 63             | ADMATCR63    | R/W | H'0000           | H'FFFE607E | 16             |
|                    | A-DMAC transfer count register 64             | ADMATCR64    | R/W | H'0000           | H'FFFE6080 | 16             |
|                    | A-DMAC transfer count register 65             | ADMATCR65    | R/W | H'0000           | H'FFFE6082 | 16             |
|                    | A-DMAC transfer count register 66             | ADMATCR66    | R/W | H'0000           | H'FFFE6084 | 16             |
|                    | A-DMAC transfer count register 67             | ADMATCR67    | R/W | H'0000           | H'FFFE6086 | 16             |
|                    | A-DMAC transfer count register 68             | ADMATCR68    | R/W | H'0000           | H'FFFE6088 | 16             |
|                    | A-DMAC transfer count register 69             | ADMATCR69    | R/W | H'0000           | H'FFFE608A | 16             |
|                    | A-DMAC transfer count register 70             | ADMATCR70    | R/W | H'0000           | H'FFFE608C | 16             |

| Channel            | Register Name                          | Abbreviation | R/W | Initial<br>Value | Address    | Access<br>Size |
|--------------------|--|--------------|-----|------------------|------------|----------------|
| Registers for each | A-DMAC transfer count register 71      | ADMATCR71    | R/W | H'0000           | H'FFFE608E | 16             |
| channel            | A-DMAC alias pointer register 0        | ADMAAR0      | R/W | H'0000           | H'FFFE6090 | 16             |
|                    | A-DMAC reload alias pointer register 0 | ADMARAR0     | R/W | H'0000           | H'FFFE6092 | 16             |
|                    | A-DMAC alias pointer register 1        | ADMAAR1      | R/W | H'0000           | H'FFFE6094 | 16             |
|                    | A-DMAC reload alias pointer register 1 | ADMARAR1     | R/W | H'0000           | H'FFFE6096 | 16             |
|                    | A-DMAC alias pointer register 56       | ADMAAR56     | R/W | H'0000           | H'FFFE60A0 | 16             |
|                    | A-DMAC alias pointer register 57       | ADMAAR57     | R/W | H'0000           | H'FFFE60A2 | 16             |
|                    | A-DMAC alias pointer register 58       | ADMAAR58     | R/W | H'0000           | H'FFFE60A4 | 16             |
|                    | A-DMAC alias pointer register 59       | ADMAAR59     | R/W | H'0000           | H'FFFE60A6 | 16             |
|                    | A-DMAC alias pointer register 60       | ADMAAR60     | R/W | H'0000           | H'FFFE60A8 | 16             |
|                    | A-DMAC alias pointer register 61       | ADMAAR61     | R/W | H'0000           | H'FFFE60AA | 16             |
|                    | A-DMAC alias pointer register 62       | ADMAAR62     | R/W | H'0000           | H'FFFE60AC | 16             |
|                    | A-DMAC alias pointer register 63       | ADMAAR63     | R/W | H'0000           | H'FFFE60AE | 16             |
|                    | A-DMAC alias pointer register 64       | ADMAAR64     | R/W | H'0000           | H'FFFE60B0 | 16             |
|                    | A-DMAC alias pointer register 65       | ADMAAR65     | R/W | H'0000           | H'FFFE60B2 | 16             |
|                    | A-DMAC alias pointer register 66       | ADMAAR66     | R/W | H'0000           | H'FFFE60B4 | 16             |
|                    | A-DMAC alias pointer register 67       | ADMAAR67     | R/W | H'0000           | H'FFFE60B6 | 16             |
|                    | A-DMAC alias pointer register 68       | ADMAAR68     | R/W | H'0000           | H'FFFE60B8 | 16             |

| Channel            | Register Name                    | Abbreviation | R/W                 | Initial<br>Value | Address    | Access<br>Size |
|--------------------|----------------------------------|--------------|---------------------|------------------|------------|----------------|
| Registers for each | A-DMAC alias pointer register 69 | ADMAAR69     | R/W                 | H'0000           | H'FFFE60BA | 16             |
| channel            | A-DMAC alias pointer register 70 | ADMAAR70     | R/W                 | H'0000           | H'FFFE60BC | 16             |
|                    | A-DMAC alias pointer register 71 | ADMAAR71     | R/W                 | H'0000           | H'FFFE60BE | 16             |
|                    | A-DMAC buffer register 2         | ADMABUF2     | R                   | H'0000000        | H'FFFE60C0 | 32             |
|                    | A-DMAC buffer register 3         | ADMABUF3     | R                   | H'0000000        | H'FFFE60C4 | 32             |
|                    | A-DMAC buffer register 4         | ADMABUF4     | R                   | H'0000000        | H'FFFE60C8 | 32             |
|                    | A-DMAC buffer register 5         | ADMABUF5     | R                   | H'0000000        | H'FFFE60CC | 32             |
|                    | A-DMAC buffer register 6         | ADMABUF6     | R                   | H'0000000        | H'FFFE60D0 | 32             |
|                    | A-DMAC buffer register 7         | ADMABUF7     | R                   | H'0000000        | H'FFFE60D4 | 32             |
|                    | A-DMAC receive wait register 0   | ADMARVPR0    | R/(W)* <sup>2</sup> | H'0000           | H'FFFE60E0 | 8, 16          |
|                    | A-DMAC receive wait register 1   | ADMARVPR1    | R/(W)* <sup>2</sup> | H'0000           | H'FFFE60E2 | 8, 16          |
|                    | A-DMAC receive wait register 2   | ADMARVPR2    | R/(W)* <sup>2</sup> | H'0000           | H'FFFE60E4 | 8, 16          |
|                    | A-DMAC receive wait register 3   | ADMARVPR3    | R/(W)* <sup>2</sup> | H'0000           | H'FFFE60E6 | 8, 16          |
|                    | A-DMAC receive wait register 4   | ADMARVPR4    | R/(W)* <sup>2</sup> | H'0000           | H'FFFE60E8 | 8, 16          |
|                    | A-DMAC receive wait register 5   | ADMARVPR5    | R/(W)* <sup>2</sup> | H'0000           | H'FFFE60EA | 8, 16          |
|                    | A-DMAC transmit wait register 0  | ADMATVPR0    | R/W                 | H'0000           | H'FFFE60F0 | 8, 16          |
|                    | A-DMAC transmit wait register 1  | ADMATVPR1    | R/W                 | H'0000           | H'FFFE60F2 | 8, 16          |
|                    | A-DMAC transmit wait register 2  | ADMATVPR2    | R/W                 | H'0000           | H'FFFE60F4 | 8, 16          |

Apr 01, 2014

| Channel                          | Register Name                   | Abbreviation | R/W | Initial<br>Value | Address    | Access<br>Size |
|----------------------------------|---------------------------------|--------------|-----|------------------|------------|----------------|
| Registers<br>for each<br>channel | A-DMAC transmit wait register 3 | ADMATVPR3    | R/W | H'0000           | H'FFFE60F6 | 8, 16          |
|                                  | A-DMAC transmit wait register 4 | ADMATVPR4    | R/W | H'0000           | H'FFFE60F8 | 8, 16          |
|                                  | A-DMAC transmit wait register 5 | ADMATVPR5    | R/W | H'0000           | H'FFFE60FA | 8, 16          |

Notes: 1. Writing a 0 after reading a 1 is only allowed to clear the flag.

2. Writing a 1 is only allowed to clear the flag.

# 12.2.1 A-DMAC Operation Register (ADMAOR)

ADMAOR is an 8-bit readable/writable register that specifies operation for all channels.

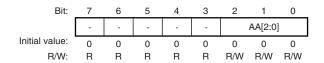
| Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0   |
|----------------|---|---|---|---|---|---|---|-----|
|                | - | - | - | - | - | - | - | DME |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   |
| R/W:           | R | R | R | R | R | R | R | R/W |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 1 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 0      | DME      | 0                | R/W | DMA master enable flag   |
|        |          |                  |     | Enables or disables DMA transfer for all channels. If this bit and the DE bit in each channel are set to 1, DMA transfer is enabled. |
|        |          |                  |     | If this bit is cleared to 0, DMA transfer is cancelled for all channels.   |

Note: If an NMI occurs, the A-DMAC enters the DMA-transfer-disabled state (for details, refer to section 12.3.3, Transfer Suspension and Resumption). To restart DMA transfer, use the NMIF bit in the DMA operation flag bit register (DMAFR) in the DMAC because the A-DMAC does not support such a bit in the ADMAOR register (for details, refer to section 11, Direct Memory Access Controller (DMAC)). Clearing the NMIF bit in the DMAC enables DMA transfer in the A-DMAC.

# 12.2.2 A-DMAC Alias Base Register (ADMAABR)

ADMAABR is an 8-bit readable/writable register that specifies the start address of the alias area (on-chip RAM are used for transfer for on-chip modules) for the A-DMAC. The alias area can be specified in 32-Kbyte units.



|        |          | Initial |     |  |
|--------|----------|---------|-----|--|
| Bit    | Bit Name | Value   | R/W | Description  |
| 7 to 3 | _        | All 0   | R   | Reserved   |
|        |          |         |     | These bits are always read as 0. The write value should always be 0. |
| 2 to 0 | AA[2:0]  | 000     | R/W | Alias area address   |
|        |          |         |     | These bits specify the start address of the alias area.              |
|        |          |         |     | 000: H'FFF80000  |
|        |          |         |     | 001: H'FFF88000  |
|        |          |         |     | 010: H'FFF90000  |
|        |          |         |     | 011: H'FFF98000  |
|        |          |         |     | 100: H'FFFA0000  |
|        |          |         |     | 101: H'FFFA8000  |
|        |          |         |     | 110: H'FFFB0000  |
|        |          |         |     | 111: H'FFFB8000  |

# 12.2.3 A-DMAC Interrupt Control Registers (ADMAIE)

ADMAIE is an 8-bit readable/writable register.

Each bit (IE bit) in each ADMAIE register enables or disables an interrupt to the CPU in each channel. If the corresponding DV bit or TE bit is set while the IE bit is set to 1, a transfer end interrupt is requested (for details, refer to section 12.3.6, Transfer Enable/Disable Conditions and Interrupt Requests).

Table 12.3 shows the correspondence between channels and bits.

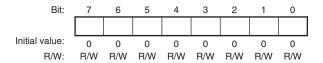


Table 12.3 Correspondence between Channels and ADMAIE Registers

| Register<br>Name | Bit        |            |            |                      |                     |            |            |            |  |  |  |  |  |
|------------------|------------|------------|------------|----------------------|---------------------|------------|------------|------------|--|--|--|--|--|
|                  | 7          | 6          | 5          | 4                    | 3                   | 2          | 1          | 0          |  |  |  |  |  |
| ADMAIE0          | Channel 7  | Channel 6  | Channel 5  | Channel 4            | Channel 4 Channel 3 |            | Reserved   | Reserved   |  |  |  |  |  |
| ADMAIE1          | Channel 15 | Channel 14 | Channel 13 | Channel 12           | Channel 11          | Channel 10 | Channel 9  | Channel 8  |  |  |  |  |  |
| ADMAIE2          | Channel 23 | Channel 22 | Channel 21 | Channel 20           | Channel 19          | Channel 18 | Channel 17 | Channel 16 |  |  |  |  |  |
| ADMAIE3          | Channel 31 | Channel 30 | Channel 29 | Channel 28           | Channel 27          | Channel 26 | Channel 25 | Channel 24 |  |  |  |  |  |
| ADMAIE4          | Channel 39 | Channel 38 | Channel 37 | Channel 36           | Channel 35          | Channel 34 | Channel 33 | Channel 32 |  |  |  |  |  |
| ADMAIE5          | Channel 47 | Channel 46 | Channel 45 | Channel 44           | Channel 43          | Channel 42 | Channel 41 | Channel 40 |  |  |  |  |  |
| ADMAIE7          | Channel 63 | Channel 62 | Channel 61 | Channel 60           | Channel 59          | Channel 58 | Channel 57 | Channel 56 |  |  |  |  |  |
| ADMAIE8          | Channel 71 | Channel 70 | Channel 69 | Channel 68 Channel 6 |                     | Channel 66 | Channel 65 | Channel 64 |  |  |  |  |  |
| ADMAIE9          |            |            | Reserved   |                      |                     | Channel 74 | Reserved   | Reserved   |  |  |  |  |  |

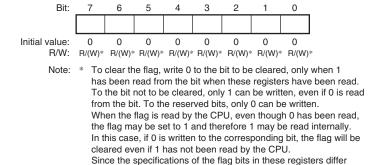
## 12.2.4 A-DMAC Data Valid Registers (ADMADV)

ADMADV is an 8-bit readable/writable register.

Each bit (DV bit) in each ADMADV register indicates the DMA transfer state of the corresponding channel. The DV bit is set to 1 to indicate that the data in the alias area is valid when a DMA transfer is completed after the DE bit is set.

Writing 1 to the DV bit is invalid. To clear the DV bit, read 1 from the DV bit and then write 0.

Table 12.4 shows the correspondence between channels and bits.



of the flags by the CPU must be executed carefully.

from the ones of the flags in other modules, the clearing operations

Table 12.4 Correspondence between Channels and ADMADV Registers

| Register | Bit        |            |            |            |            |            |            |            |  |  |  |  |  |  |
|----------|------------|------------|------------|------------|------------|------------|------------|------------|--|--|--|--|--|--|
| Name     | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |  |  |  |  |  |  |
| ADMADV0  | Channel 7  | Channel 6  | Channel 5  | Channel 4  | Channel 3  | Channel 2  | Reserved   | Reserved   |  |  |  |  |  |  |
| ADMADV1  | Channel 15 | Channel 14 | Channel 13 | Channel 12 | Channel 11 | Channel 10 | Channel 9  | Channel 8  |  |  |  |  |  |  |
| ADMADV2  | Channel 23 | Channel 22 | Channel 21 | Channel 20 | Channel 19 | Channel 18 | Channel 17 | Channel 16 |  |  |  |  |  |  |
| ADMADV3  | Channel 31 | Channel 30 | Channel 29 | Channel 28 | Channel 27 | Channel 26 | Channel 25 | Channel 24 |  |  |  |  |  |  |
| ADMADV4  | Channel 39 | Channel 38 | Channel 37 | Channel 36 | Channel 35 | Channel 34 | Channel 33 | Channel 32 |  |  |  |  |  |  |
| ADMADV5  | Channel 47 | Channel 46 | Channel 45 | Channel 44 | Channel 43 | Channel 42 | Channel 41 | Channel 40 |  |  |  |  |  |  |

## 12.2.5 A-DMAC Transfer End Registers (ADMATE)

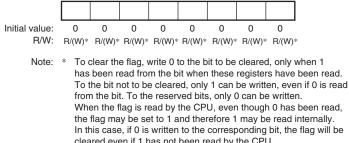
ADMATE is an 8-bit readable/writable register.

Bit:

Each bit (TE bit) in each ADMATE register indicates the DMA transfer state of the corresponding channel. The TE bit is set to 1 when the specified counts of DMA transfers are completed.

Writing 1 to the TE bit is invalid. To clear the TE bit, read 1 from the TE bit and then write 0 to it. While the TE bit is set to 1, the corresponding channel cannot accept the transfer request.

Table 12.5 shows the correspondence between channels and ADMATE registers.



cleared even if 1 has not been read by the CPU. Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

Table 12.5 Correspondence between Channels and ADMATE Registers

| Register | Bit        |            |            |            |            |            |            |            |  |  |  |  |  |
|----------|------------|------------|------------|------------|------------|------------|------------|------------|--|--|--|--|--|
| Name     | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |  |  |  |  |  |
| ADMATE0  | Channel 63 | Channel 62 | Channel 61 | Channel 60 | Channel 59 | Channel 58 | Channel 57 | Channel 56 |  |  |  |  |  |
| ADMATE1  | Channel 71 | Channel 70 | Channel 69 | Channel 68 | Channel 67 | Channel 66 | Channel 65 | Channel 64 |  |  |  |  |  |
| ADMATE2  |            |            | Reserved   |            |            | Channel 74 | Reserved   | Reserved   |  |  |  |  |  |

# 12.2.6 A-DMAC Enable Registers (ADMADE)

ADMADE is an 8-bit readable/writable register.

Each bit (DE bit) in each ADMADE register enables or disables DMA transfer for a channel for ATU-III (timers A, C, and F) or RCAN. Setting the DE bit of the corresponding channel and the DME bit in ADMAOR to 1 enables DMA transfer.

Table 12.6 shows the correspondence between channels and bits.

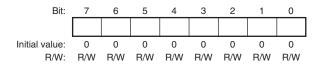


Table 12.6 Correspondence between Channels and ADMADE Registers

| Register | Bit                |            |            |                     |            |                       |            |            |  |  |  |  |  |
|----------|--------------------|------------|------------|---------------------|------------|-----------------------|------------|------------|--|--|--|--|--|
| Name     | 7                  | 6          | 5          | 4                   | 3          | 2                     | 1          | 0          |  |  |  |  |  |
| ADMADE0  | Channel 7          | Channel 6  | Channel 5  | Channel 4 Channel 3 |            | Channel 2             | Reserved   | Reserved   |  |  |  |  |  |
| ADMADE1  | Channel 15         | Channel 14 | Channel 13 | Channel 12          | Channel 11 | Channel 11 Channel 10 |            | Channel 8  |  |  |  |  |  |
| ADMADE2  | Channel 23         | Channel 22 | Channel 21 | Channel 20          | Channel 19 | nannel 19 Channel 18  |            | Channel 16 |  |  |  |  |  |
| ADMADE3  | Channel 31         | Channel 30 | Channel 29 | Channel 28          | Channel 27 | Channel 26            | Channel 25 | Channel 24 |  |  |  |  |  |
| ADMADE4  | Channel 39         | Channel 38 | Channel 37 | Channel 36          | Channel 35 | Channel 34            | Channel 33 | Channel 32 |  |  |  |  |  |
| ADMADE5  | Channel 47 Channel |            | Channel 45 | Channel 44          | Channel 43 | Channel 42            | Channel 41 | Channel 40 |  |  |  |  |  |
| ADMADE7  |                    |            | Reserved   |                     |            | Channel 74            | Reserved   | Channel 72 |  |  |  |  |  |

# 12.2.7 A-DMAC Transfer Mode Registers (ADMAMODE)

ADMAMODE is an 8-bit readable/writable register.

Each bit (MODE bit) in each ADMAMODE register specifies the transfer mode for each channel used for ATM-III (timer F). A write to this register should be performed in the transfer-disabled state (DE = 0 or DME = 0).

To use timer F in PWM input waveform measurement mode, set the MODE bit to 1; otherwise, clear the MODE bit to 0.

Table 12.7 shows the correspondence between channels and the ADMAMODE register.

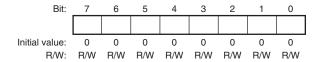


Table 12.7 Correspondence between Channels and ADMAMODE Registers

| Register  | Bit        |            |            |            |            |            |            |            |  |  |  |  |  |
|-----------|------------|------------|------------|------------|------------|------------|------------|------------|--|--|--|--|--|
| Name      | 7          | 6          | 5 4        |            | 3 2        |            | 1          | 0          |  |  |  |  |  |
| ADMAMODE0 | Channel 31 | Channel 30 | Channel 29 | Channel 28 | Reserved   | Reserved   | Reserved   | Reserved   |  |  |  |  |  |
| ADMAMODE1 | Channel 39 | Channel 38 | Channel 37 | Channel 36 | Channel 35 | Channel 4  | Channel 33 | Channel 32 |  |  |  |  |  |
| ADMAMODE2 | Channel 47 | Channel 46 | Channel 45 | Channel 44 | Channel 43 | Channel 42 | Channel 41 | Channel 40 |  |  |  |  |  |

## 12.2.8 A-DMAC Transfer Count Registers (ADMATCR)

ADMATCR is a 16-bit readable/writable register that specifies the DMA transfer count for channels used for ATU-III (timer G), ADC, RSPI, and SCI.

While DMA transfer is enabled, the ADMATCR value is updated (post-decremented) each time a DMA transfer is performed, thus indicating the remaining transfer count till transfer end. The ADMATCR registers can be specified from H'0000 to H'03FF.

Clearing ADMATCR to 0 suspends the DMA transfer. Note that writing a value other than 0 to ADMATCR while ADMATCR  $\neq$  0 is prohibited.

The ADMARTCR value is reloaded into ADMATCR corresponding to channels for ATU-III (timer G) and ADC, if a DMA transfer is performed while ADMATCR = 1 and ADMARTCR  $\neq$  0.

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

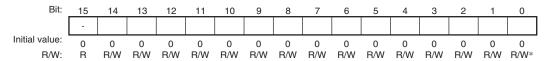
#### 12.2.9 A-DMAC Alias Pointer Registers (ADMAAR)

ADMAAR is a 16-bit readable/writable register that specifies the alias area for channels used for ATU-III (timer G), ADC, RSPI, and SCI. The ADMAAR value is updated (post-updated) to indicate the next transfer address each time a DMA transfer is performed. ADMAAR must be written while the corresponding ADMATCR is cleared to 0.

The alias area for each channel is defined as an area starting from the address specified by ADMAAR to the address obtained by the transfer count specified by ADMATCR × 2 bytes (for channels used for RSPI and ADC) or to the address obtained by the transfer count specified by ADMATCR × 1 byte (for channels used for the SCI and ATU-III (timer G)).

The transfer destination address is specified by the relative address from the start address of the alias area. The specifiable relative address is from H'0000 to H'7FFF.

The ADMARAR value will be reloaded into ADMAAR corresponding to channels for ATU-III (timer G) and the ADC, if a DMA transfer is performed while ADMATCR = 1 and ADMARTCR  $\neq 0$ .



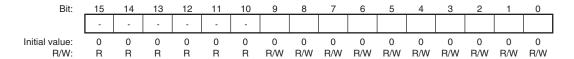
A writing to bit 0 in the ADMAAR registers for channels used for ADC and RSPI is invalid.

# 12.2.10 A-DMAC Reload Transfer Count Registers (ADMARTCR)

ADMARTCR is a 16-bit readable/writable register that specifies reload values of the corresponding ADMATCR in a channel used for ADC. ADMARTCR can be specified from H'0000 to H'03FF.

The ADMARTCR value will be reloaded into ADMATCR if a DMA transfer is performed while ADMATCR = 1 and ADMARTCR  $\neq$  0.

ADMARTCR must be set before ADMATCR is set.

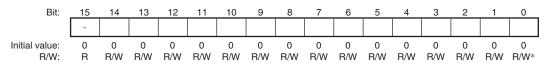


#### 12.2.11 A-DMAC Reload Alias Pointer Registers (ADMARAR)

ADMARAR is a 16-bit readable/writable register that specifies reload values of the corresponding ADMAAR in a channel used for ADC. ADMARAR can be specified from H'0000 to H'7FFF.

The ADMARAR value is reloaded into ADMAAR if a DMA transfer is performed while ADMATCR = 1 and ADMARTCR  $\neq$  0.

ADMARAR must be set before the ADMAAR registers are set.



Note: \* A writing to bit 0 in the ADMARAR registers for channels used for ADC is invalid.

## **A-DMAC Buffer Registers (ADMABUF)**

ADMABUF is a 32-bit read-only register. ADMABUF2 to ADMABUF7 correspond to channels 2 to 7 for timer A.

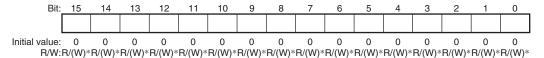
ADMABUF stores the value read for the input capture register according to the transfer request from timer A. Two values: data stored in ADMABUF (value read for the input capture register for the previous transfer request from timer A) and the value read for the input capture register from timer A will be transferred to the alias area using the timer A transfer request as a trigger.

| Bit:                   | 31     | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     |
|------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|                        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Initial value:         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |
| R/W:                   | R      | R      | R      | R      | R      | R      | R      | R      | R      | R      | R      | R      | R      | R      | R      | R      |
| Bit:                   | 15     | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|                        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Initial value:<br>R/W: | 0<br>R | 0<br>B | 0<br>R | 0<br>R | 0<br>R | 0<br>R | 0<br>B |

#### 12.2.13 A-DMAC Receive Wait Registers (ADMARVPR)

ADMARVPR is a 16-bit readable/writable register that includes the flags corresponding to mailboxes (MBx) of the RCAN module. Only 1 can be written to ADMARVPR to clear flags in ADMARVPR.

Table 12.8 shows the correspondence between ADMARVPR and mailboxes in the RCAN.



Note: \* To clear the flag, write 1 to the bit to be cleared, only when 1 has been read from the bit when these registers have been read.

To the bit not to be cleared, only 0 can be written, even if 0 is read from the bit. To the reserved bits, only 0 can be written.

When the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally. In this case, if 1 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU.

Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

Table 12.8 Correspondence between ADMARVPR and Mailboxes in RCAN

| ADMARVPR  | Bits    | RCAN Module | MBx      |
|-----------|---------|-------------|----------|
| ADMARVPR0 | 15 to 0 | RCAN_A      | 31 to 16 |
| ADMARVPR1 | 15 to 0 |             | 15 to 0  |
| ADMARVPR2 | 15 to 0 | RCAN_B      | 31 to 16 |
| ADMARVPR3 | 15 to 0 |             | 15 to 0  |
| ADMARVPR4 | 15 to 0 | RCAN_C      | 31 to 16 |
| ADMARVPR5 | 15 to 0 |             | 15 to 0  |

A flag in ADMARVPR corresponding to the mailbox in the RCAN module is set to 1 when data in the mailbox in the RCAN module has been transferred to the alias area according to the transfer request from the RCAN. To clear a flag (RV bit) in ADMAPVPR, read 1 from the flag (RV bit) and then write 1 to it again.

#### 12.2.14 A-DMAC Transmit Wait Registers (ADMATVPR)

ADMATVPR is a 16-bit readable/writable register that includes the flags corresponding to mailboxes (MBx) of the RCAN module.

ADMATVPR specifies the IDs of mailboxes, to which data is transferred from the alias area, by software. If a bit corresponding to the mailbox (MBx) is set to 1, data is transferred from the alias area to the mailbox (MBx) in transfer enable state (DME = 1, DE = 1, and TE = 0).

When data transfer is completed, the flag corresponding to the mailbox (MBx) is cleared to 0 by hardware.

A bit corresponding to a mailbox (MBx), which is not specified as a transmitter mailbox in the RCAN modules, cannot be set to 1. Note that all the ADMATVPR registers cannot be written to while the RCAN transmitter channel is in transfer enable state or in suspend state (for details, refer to section 12.3.3, Transfer Suspension and Resumption).

Table 12.9 shows the correspondence between ADMATVPR registers and mailboxes in RCAN modules.

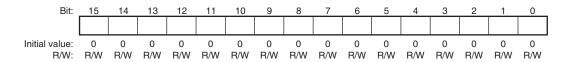


Table 12.9 Correspondence between ADMATVPR Registers and Mailboxes in RCAN Modules

| ADMARVPR  | Bits     | RCAN Module | MBx      |
|-----------|----------|-------------|----------|
| ADMATVPR0 | 15 to 0  | RCAN_A      | 31 to 16 |
| ADMATVPR1 | 15 to 1* |             | 15 to 1* |
| ADMATVPR2 | 15 to 0  | RCAN_B      | 31 to 16 |
| ADMATVPR3 | 15 to 1* |             | 15 to 1* |
| ADMATVPR4 | 15 to 0  | RCAN_C      | 31 to 16 |
| ADMATVPR5 | 15 to 1* |             | 15 to 1* |

Note: \* MB0 cannot be specified because the MB0 is a mailbox for reception. Accordingly, a write to bit 0 of the corresponding ADMATVTR register is invalid.

# 12.3 Operation

#### 12.3.1 Alias Areas

Areas in on-chip RAM that can be used as a transfer source and destination for A-DMAC data transfer are called alias areas.

The start address of an alias area can be specified by ADMAABR in 32-Kbyte units. Figure 12.2 shows the memory map of the alias areas (ADMAABR is specified as default value).

In channels for ATU-III (timers A, C, and F) and RCAN, the relative address corresponding to the address specified by ADMAABR for each channel alias area is fixed. For details, refer to section 12.3.2 (2), Operation for A-DMAC Channels Used for ATU-III (timers A, C, and F) and section 12.3.2 (4), Operation for A-DMAC Channels Used for RCAN.

While in channels for ATU-III (timer G), ADC, RSPI, and SCI, the alias area for each channel can be specified. The start address and area size of each alias area are specified by the ADMAAR and ADMATCR registers, respectively. For details, refer to section 12.2.8, A-DMAC Transfer Count Registers (ADMATCR) and section 12.2.9, A-DMAC Alias Pointer Registers (ADMAAR).

Note: Illegal settings such as alias area overlap and alias area setting exceeding the on-chip RAM maximum address can be specified according to the ADMAAR and ADMATCR settings. These illegal settings cannot be detected by hardware.

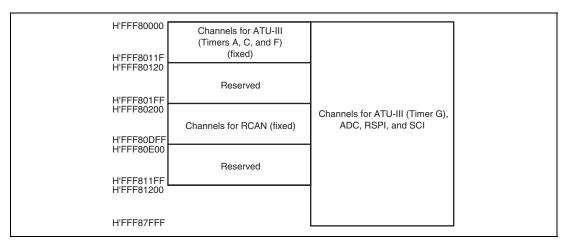


Figure 12.2 Memory Map of Alias Areas

## 12.3.2 Each A-DMAC Channel Operation

#### (1) Operation for A-DMAC Channels Used for ATU-III (Timer G) and ADC

#### Overview

A-DMAC channel (channel 0) for the ATU-III (timer G) supports the ADC input signal switch setting via a port. This channel performs data transfer from the alias area to port G that controls the ADC input using a transfer request from timer G as a source.

A-DMAC channel (channel 1) for ADC supports the reading of A/D conversion result from the ADC. This channel performs data transfer from the ADR0 register that stores the A/D conversion result of the ADC to the alias area.

Alias areas for ATU-III (timer G) and ADC channels can be specified by ADMAAR and ADMATCR.

Figure 12.3 shows an overview of the DMA transfer.

- Transfer request
  - Channel 0: Timer G2 compare match
  - Channel 1: ADC\_A and AN0 interrupt conversion end of ADC

#### Addressing

Addressing in registers is fixed (not incremented or decremented). Addressing in alias area is post-incremented (incremented by one in channel 0 and incremented by two in channel 1) each transfer.

If the next transfer is performed while ADMATCR = 1 and ADMARTCR  $\neq$  0, ADMATCR and ADMAAR are updated by the corresponding reload register values. Accordingly, by setting ADMAAR = ADMARAR and ADMATCR = ADMARTCR, the data structure of the ring list (ring buffer) can be configured in the alias area.

#### Transfer flow

Data transfer is performed in the following sequence. Figure 12.4 shows the flowchart of data transfer.

- 1. Sets reload registers for ADMATCR and ADMAAR.
- 2. Specifies the start address of the alias area by ADMAAR and specifies the transfer count by ADMATCR.
- 3. Performs one transfer if a transfer request occurs while transfer is enabled (DME = 1, NMIF = 0, ADMATCR ≠ 0), decrements ADMATCR, and updates ADMAAR.
- 4. Transfers reload register values to ADMATCR and ADMAAR if data transfer is performed while ADMATCR = 1 and ADMARTCR ≠ 0.

Suspends data transfer if an NMI interrupt occurs or if the DME bit is cleared to 0.
 Disables data transfer if 0 is written to ADMATCR while ADMATCR ≠ 0.

#### Notes

Reload registers ADMARAR and ADMARTCR for ADMAAR and ADMATCR must be set before ADMAAR and ADMATCR are set. Otherwise, reloading may not be performed correctly.

In the alias area used as a transfer source of channel 0, the data for ADC input signal switching must be specified by the upper 4 bits in each byte.

In the alias area used for channel 1, the contents in the alias area used as a transfer destination are updated with the address register whenever a transfer is requested. Accordingly, the contents of the alias area will be overwritten before they are read.

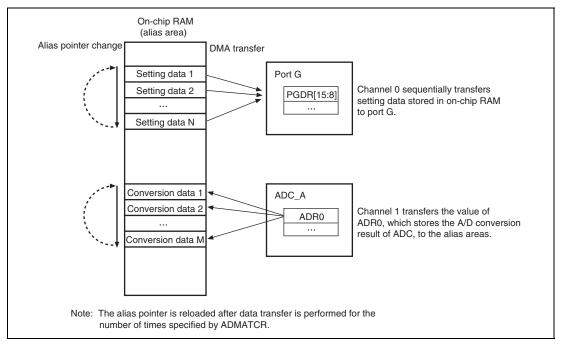


Figure 12.3 Transfer Overview (Channels for ATU-III (timer G) and ADC)

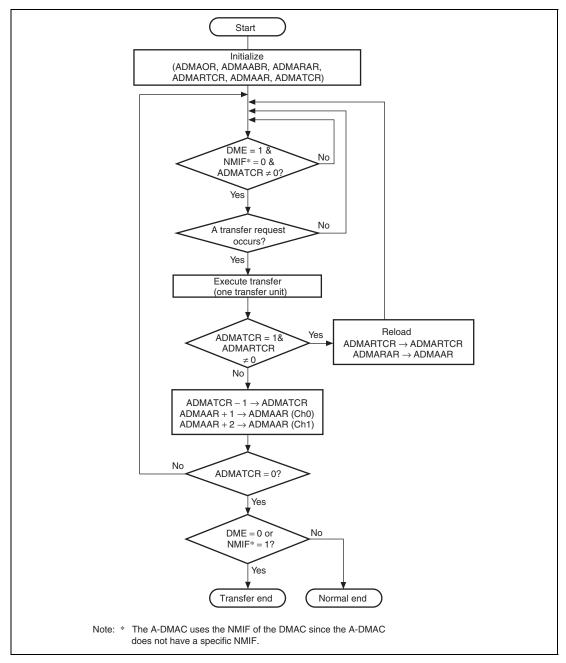


Figure 12.4 Flowchart of DMA Transfer (Channels for ATU-III (timer G) and ADC)

#### (2) Operation for A-DMAC Channels Used for ATU-III (timers A, C, and F)

#### Overview

A-DMAC channels for the ATU-III (timers A, C, and F) perform data transfer from ATU-III registers to the alias areas. The alias area size for each channel is defined as follows.

- A-DMAC channel for timer A: 8 bytes (area for 32-bit register  $\times$  2)
- A-DMAC channel for timer C: 4 bytes
- A-DMAC channel for timer F: 8 bytes (area for 32-bit register  $\times$  2)

Refer to table 12.10, for details on the relationship between transfer source register name and transfer destination address. Figure 12.5 shows an overview of the DMA transfer.

## Transfer request

An input capture interrupt of the corresponding ATU-III channel is used as a transfer request.

## Addressing

Both transfer source and destination addresses are fixed (not incremented or decremented).

#### Transfer flow

Data transfer is performed in the following sequence. Figure 12.6 shows the flowchart of data transfer.

- 1. Performs one transfer if a transfer request occurs while transfer is enabled (DE =1, DME = 1, NMIF = 0), and sets the DV bit to 1. In this case, if an interrupt is enabled (IE = 1), requests a transfer end interrupt to the CPU.
- 2. Performs one transfer each time a transfer request occurs while transfer is enabled.
- 3. Disables transfer if an NMI interrupt occurs, or if the DME or DE bit is cleared to 0.

#### Notes

In transfer enable state, the alias area specified as transfer destination is overwritten each time a transfer is requested and the alias area always indicates the latest register value.

In A-DMAC channel for timer A, input capture value to be read for a current transfer request and ADMABUF register storage value (= input capture value read in the previous transfer request) are transferred to the alias area. (For details, refer to section 12.3.4 (1), Supplementary Description for A-DMAC Channels Used for ATU-III (timer A).)

In A-DMAC channel for timer F, registers to be transferred differ depending on the corresponding ADMAMODE settings. If the ADMAMODE bit is cleared to 0 (default), the CDRF registers are transferred to the upper 4 bytes of alias area for each channel. While, if the ADMAMODE bit is set to 1, the CDRF and GRCF registers are transferred to the upper and lower 4 bytes of each alias area, respectively.

Table 12.10 A-DMAC Channels for ATU-III (Timers A, C, and F)

| Channel<br>No. | Transfer  | Request Sourc | e:e       | Transfer Register | Relative Addresses in Alias Area |
|----------------|-----------|---------------|-----------|-------------------|----------------------------------|
| 2              | ATU-III   | Channel 0     |           | ICRA0             | H'000 to H'007                   |
| 3              | (Timer A) | Channel 1     |           | ICRA1             | H'008 to H'00F                   |
| 4              | _         | Channel 2     |           | ICRA2             | H'010 to H'017                   |
| 5              | =         | Channel 3     |           | ICRA3             | H'018 to H'01F                   |
| 6              | =         | Channel 4     |           | ICRA4             | H'020 to H'027                   |
| 7              | =         | Channel 5     |           | ICRA5             | H'028 to H'02F                   |
| 8              | ATU-III   | Subblock C0   | Channel 0 | GRC00             | H'030 to H'033                   |
| 9              | (Timer C) |               | Channel 1 | GRC01             | H'034 to H'037                   |
| 10             | =         |               | Channel 2 | GRC02             | H'038 to H'03B                   |
| 11             | _         |               | Channel 3 | GRC03             | H'03C to H'03F                   |
| 12             | =         | Subblock C1   | Channel 0 | GRC10             | H'040 to H'043                   |
| 13             | =         |               | Channel 1 | GRC11             | H'044 to H'047                   |
| 14             | =         |               | Channel 2 | GRC12             | H'048 to H'04B                   |
| 15             | =         |               | Channel 3 | GRC13             | H'04C to H'04F                   |
| 16             | =         | Subblock C2   | Channel 0 | GRC20             | H'050 to H'053                   |
| 17             | _         |               | Channel 1 | GRC21             | H'054 to H'057                   |
| 18             | =         |               | Channel 2 | GRC22             | H'058 to H'05B                   |
| 19             | =         |               | Channel 3 | GRC23             | H'05C to H'05F                   |
| 20             | _         | Subblock C3   | Channel 0 | GRC30             | H'060 to H'063                   |
| 21             | =         |               | Channel 1 | GRC31             | H'064 to H'067                   |
| 22             | _         |               | Channel 2 | GRC32             | H'068 to H'06B                   |
| 23             | =         |               | Channel 3 | GRC33             | H'06C to H'06F                   |
| 24             | _         | Subblock C4   | Channel 0 | GRC40             | H'070 to H'073                   |
| 25             | _         |               | Channel 1 | GRC41             | H'074 to H'077                   |
| 26             | =         |               | Channel 2 | GRC42             | H'078 to H'07B                   |
| 27             | _         |               | Channel 3 | GRC43             | H'07C to H'07F                   |

| Channel<br>No. | Transfer  | Request Source | Transfer Register | Relative Addresses in Alias Area |
|----------------|-----------|----------------|-------------------|----------------------------------|
| 28             | ATU-III   | Channel 0      | CDRF00, GRCF00    | H'080 to H'087                   |
| 29             | (Timer F) | Channel 1      | CDRF01, GRCF01    | H'088 to H'08F                   |
| 30             | _         | Channel 2      | CDRF02, GRCF02    | H'090 to H'097                   |
| 31             | _         | Channel 3      | CDRF03, GRCF03    | H'098 to H'09F                   |
| 32             | _         | Channel 4      | CDRF04, GRCF04    | H'0A0 to H'0A7                   |
| 33             | _         | Channel 5      | CDRF05, GRCF05    | H'0A8 to H'0AF                   |
| 34             | _         | Channel 6      | CDRF06, GRCF06    | H'0B0 to H'0B7                   |
| 35             | _         | Channel 7      | CDRF07, GRCF07    | H'0B8 to H'0BF                   |
| 36             | _         | Channel 8      | CDRF08, GRCF08    | H'0C0 to H'0C7                   |
| 37             | _         | Channel 9      | CDRF09, GRCF09    | H'0C8 to H'0CF                   |
| 38             | _         | Channel 10     | CDRF10, GRCF10    | H'0D0 to H'0D7                   |
| 39             | _         | Channel 11     | CDRF11, GRCF11    | H'0D8 to H'0DF                   |
| 40             | _         | Channel 12     | CDRF12, GRCF12    | H'0E0 to H'0E7                   |
| 41             | _         | Channel 13     | CDRF13, GRCF13    | H'0E8 to H'0EF                   |
| 42             | _         | Channel 14     | CDRF14, GRCF14    | H'0F0 to H'0F7                   |
| 43             | _         | Channel 15     | CDRF15, GRCF15    | H'0F8 to H'0FF                   |
| 44             | _         | Channel 16     | CDRF16, GRCF16    | H'100 to H'107                   |
| 45             | =         | Channel 17     | CDRF17, GRCF17    | H'108 to H'10F                   |
| 46             | =         | Channel 18     | CDRF18, GRCF18    | H'110 to H'117                   |
| 47             | =         | Channel 19     | CDRF19, GRCF19    | H'118 to H'11F                   |

# (3) Operation for A-DMAC Channels Used for RSPI and SCI

#### Overview

A-DMAC channels for the RSPI and SCI supports the following data transfer. In even channels, data transfer from a register to the alias area is performed; while in odd channels, data transfer from the alias area to a register is performed. Figure 12.5 shows an overview of the DMA transfer.

The alias area size for each channel is obtained by (read (write) register size) × (transfer count specified by ADMATCR). Maximum alias sizes for RSPI channels and SCI channels are 2046 bytes and 1023 bytes, respectively. (For details, refer to section 12.2.7, A-DMAC Transfer Mode Registers (ADMAMODE), and section 12.2.9, A-DMAC Alias Pointer Registers (ADMAAR).)

Table 12.11 summarizes each channel transfer request, transfer source register name, and transfer direction.

#### Transfer request

An RSPI or SCI reception or transmission interrupt for each channel is used as a transfer request. In transfer enable state, one data transfer is performed for a transfer request. In even channels for RSPI, a request generated at RSPI receive buffer full is used as a DMA transfer request. In odd channels for the RSPI, a request generated at RSPI transmit buffer empty is used as a DMA request.

In even channels for the SCI, a request generated at SCI receive data full is used as a DMA transfer request. In odd channels for the SCI, a request generated at SCI transmit data empty is used as a DMA request.

The transfer enable state is entered if the transfer count is specified in ADMATCR corresponding to each channel, if the TE bit is cleared to 0 (only when the TE bit is set to 1), and if the DME bit in ADMAOR is set to 1. (For details, refer to section 12.3.6, Transfer Enable/Disable Conditions and Interrupt Requests.)

## Addressing

Addressing in registers is fixed (not incremented or decremented). Addressing in alias area is post-incremented (incremented by two in channels for RSPI and incremented by one in channels for the SCI) each transfer.

#### Transfer flow

Data transfer is performed in the following sequence. Figure 12.6 shows the flowchart of data transfer.

- 1. Specifies the start address of the alias area by ADMAAR and specifies the transfer count by ADMATCR.
- Performs one transfer if a transfer request occurs while transfer is enabled (DME = 1, NMIF = 0, ADMATCR ≠ 0, TE = 0), decrements ADMATCR, and increments ADMAAR.
- 3. Sets the TE bit to 1 if ADMATCR is cleared 0 and if data transfer of specified count has been performed, and requests an interrupt to the CPU if an interrupt is enabled (IE = 1).
- 4. Suspends data transfer if an NMI interrupt occurs or if the DME bit is cleared to 0. Disables data transfer if 0 is written to ADMATCR while ADMATCR ≠ 0.

#### Notes

To re-enable data transfer while data transfer has been completed (TE = 1, ADMATCR = 0), ADMATCR must be set before the TE bit is cleared to 0. If the TE bit is cleared to 0 first, the transfer request may be directly informed to the interrupt controller. (For details, refer to section 12.3.6, Transfer Enable/Disable Conditions and Interrupt Requests.)

Table 12.11 Channels for RSPI and SCI

| Channel<br>No. | Transfer Request Source |           | Transfer Request Signal      | Transfer<br>Register | Transfer Direction | Transfer<br>Bytes |
|----------------|-------------------------|-----------|------------------------------|----------------------|--------------------|-------------------|
| 56             | RSPI                    | Channel A | Receive buffer full (SPRI)   | SPDRA                | IO →<br>RAM        | 2                 |
| 57             | _                       |           | Transmit buffer empty (SPTI) | SPDRA                | IO ←<br>RAM        | 2                 |
| 58             | _                       | Channel B | Receive buffer full (SPRI)   | SPDRB                | IO →<br>RAM        | 2                 |
| 59             | _                       |           | Transmit buffer empty (SPTI) | SPDRB                | IO ←<br>RAM        | 2                 |
| 60             | _                       | Channel C | Receive buffer full (SPRI)   | SPDRC                | IO →<br>RAM        | 2                 |
| 61             | _                       |           | Transmit buffer empty (SPTI) | SPDRC                | IO ←<br>RAM        | 2                 |
| 62             | SCI                     | Channel A | Receive data full            | SCRDR1A              | IO →<br>RAM        | 1                 |
| 63             | _                       |           | Transmit data empty          | SCTDR1A              | IO ←<br>RAM        | 1                 |
| 64             | _                       | Channel B | Receive data full            | SCRDR1B              | IO →<br>RAM        | 1                 |
| 65             | _                       |           | Transmit data empty          | SCTDR1B              | IO ←<br>RAM        | 1                 |
| 66             | _                       | Channel C | Receive data full            | SCRDR1C              | IO →<br>RAM        | 1                 |
| 67             | _                       |           | Transmit data empty          | SCTDR1C              | IO ←<br>RAM        | 1                 |
| 68             | _                       | Channel D | Receive data full            | SCRDR1D              | IO →<br>RAM        | 1                 |
| 69             | _                       |           | Transmit data empty          | SCTDR1D              | IO ←<br>RAM        | 1                 |
| 70             | _                       | Channel E | Receive data full            | SCRDR1E              | IO →<br>RAM        | 1                 |
| 71             | -                       |           | Transmit data empty          | SCTDR1E              | IO ←<br>RAM        | 1                 |

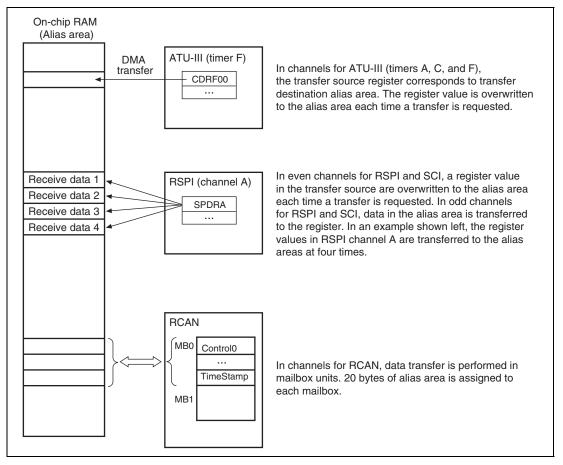


Figure 12.5 Transfer Overview (Channels for ATU-III (Timers A, C, and F), RSPI, SCI, and RCAN)

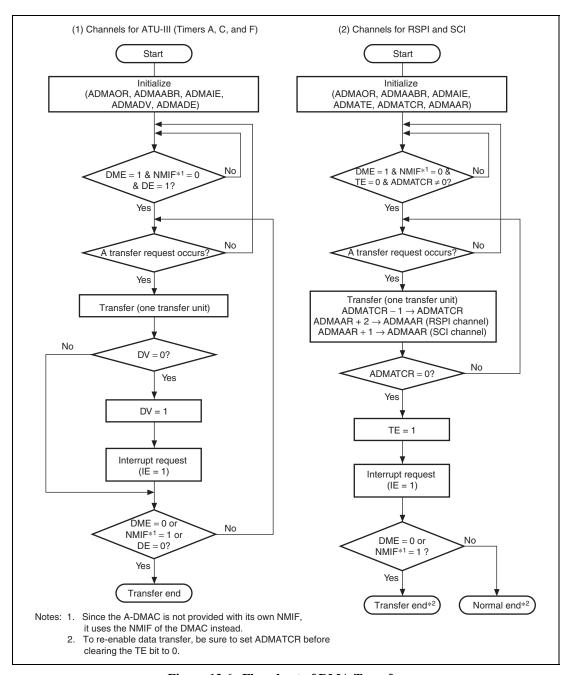


Figure 12.6 Flowchart of DMA Transfer (Channels for ATU-III (Timers A, C, and F), RSPI, and SPI)

## (4) Operation for A-DMAC Channels Used for RCAN

#### Overview

A-DMAC channels for the RCAN performs data transfer between mailboxes in the RCAN module and alias areas. In channel 72, a receive operation is performed (data is transferred from a mailbox (MBx) to the alias area). In channel 74, a transmit operation is performed (data is transferred from alias area to the mailbox (MBx)). Figure 12.5 shows an overview of the DMA transfer.

Table 12.12 shows the correspondence between mailboxes (MBx) and alias areas. The alias area for each mailbox (MBx) is assigned in 32-byte units. Receive alias areas are common to transmit alias areas.

For one mailbox (MBx) transfer, data is transferred in long-word units at five times and a total of 20-byte data is transferred. Fields to be transferred are ID (4 bytes), LAFM (4 bytes), Data (8 bytes), Control 1 (2 bytes), and TimeStamp (2 bytes).

Table 12.12 Correspondence between Mailbox (MBx) and Alias Area Addresses

| MBx | RCAN_A           | RCAN_B           | RCAN_C           |
|-----|------------------|------------------|------------------|
| 0   | H'0200 to H'0213 | H'0600 to H'0613 | H'0A00 to H'0A13 |
| 1   | H'0220 to H'0233 | H'0620 to H'0633 | H'0A20 to H'0A33 |
| 2   | H'0240 to H'0253 | H'0640 to H'0653 | H'0A40 to H'0A53 |
| 3   | H'0260 to H'0273 | H'0660 to H'0673 | H'0A60 to H'0A73 |
| 4   | H'0280 to H'0293 | H'0680 to H'0693 | H'0A80 to H'0A93 |
| 5   | H'02A0 to H'02B3 | H'06A0 to H'06B3 | H'0AA0 to H'0AB3 |
| 6   | H'02C0 to H'02D3 | H'06C0 to H'06D3 | H'0AC0 to H'0AD3 |
| 7   | H'02E0 to H'02F3 | H'06E0 to H'06F3 | H'0AE0 to H'0AF3 |
| 8   | H'0300 to H'0313 | H'0700 to H'0713 | H'0B00 to H'0B13 |
| 9   | H'0320 to H'0333 | H'0720 to H'0733 | H'0B20 to H'0B33 |
| 10  | H'0340 to H'0353 | H'0740 to H'0753 | H'0B40 to H'0B53 |
| 11  | H'0360 to H'0373 | H'0760 to H'0773 | H'0B60 to H'0B73 |
| 12  | H'0380 to H'0393 | H'0780 to H'0793 | H'0B80 to H'0B93 |
| 13  | H'03A0 to H'03B3 | H'07A0 to H'07B3 | H'0BA0 to H'0BB3 |
| 14  | H'03C0 to H'03D3 | H'07C0 to H'07D3 | H'0BC0 to H'0BD3 |
| 15  | H'03E0 to H'03F3 | H'07E0 to H'07F3 | H'0BE0 to H'0BF3 |
| 16  | H'0400 to H'0413 | H'0800 to H'0813 | H'0C00 to H'0C13 |
| 17  | H'0420 to H'0433 | H'0820 to H'0833 | H'0C20 to H'0C33 |
| 18  | H'0440 to H'0453 | H'0840 to H'0853 | H'0C40 to H'0C53 |
| 19  | H'0460 to H'0473 | H'0860 to H'0873 | H'0C60 to H'0C73 |
| 20  | H'0480 to H'0493 | H'0880 to H'0893 | H'0C80 to H'0C93 |
| 21  | H'04A0 to H'04B3 | H'08A0 to H'08B3 | H'0CA0 to H'0CB3 |
| 22  | H'04C0 to H'04D3 | H'08C0 to H'08D3 | H'0CC0 to H'0CD3 |
| 23  | H'04E0 to H'04F3 | H'08E0 to H'08F3 | H'0CE0 to H'0CF3 |
| 24  | H'0500 to H'0513 | H'0900 to H'0913 | H'0D00 to H'0D13 |
| 25  | H'0520 to H'0533 | H'0920 to H'0933 | H'0D20 to H'0D33 |
| 26  | H'0540 to H'0553 | H'0940 to H'0953 | H'0D40 to H'0D53 |
| 27  | H'0560 to H'0573 | H'0960 to H'0973 | H'0D60 to H'0D73 |
| 28  | H'0580 to H'0593 | H'0980 to H'0993 | H'0D80 to H'0D93 |
| 29  | H'05A0 to H'05B3 | H'09A0 to H'09B3 | H'0DA0 to H'0DB3 |
| 30  | H'05C0 to H'05D3 | H'09C0 to H'09D3 | H'0DC0 to H'0DD3 |
| 31  | H'05E0 to H'05F3 | H'09E0 to H'09F3 | H'0DE0 to H'0DF3 |

Note: Only the lower addresses are indicated in hexadecimal.

#### Transfer request

A channel for reception uses the MBx full information from the RCAN uses a transfer request. The mailbox (MBx) to be transferred is detected by scanning the mailbox (MBx) status.

A channel for transmission is starts transfer by software.

## Addressing

Addressing in both transfer source and destination are incremented.

Operation Details and Transfer flow

Channel 72: One mailbox (MBx) transfer is performed by a transfer request. The mailbox (MBx) number to be transferred is informed by the RCAN synchronously with the transfer request.

When the mailbox (MBx) transfer is completed, the RV bit corresponding to the mailbox (MBx) is set to 1. Even if the mailbox (MBx) transfer whose RV bit is set to 1 is requested from the RCAN, the A-DMAC does not accept the transfer request. In this case, the A-DMAC informs the RCAN of the MBx transfer end and the RCAN clears the corresponding bit in the receive wait register (or frame wait register).

Data transfer is performed in the following sequence. Figure 12.7 shows the flowchart of data transfer.

- A. Performs one mailbox (MBx) transfer if a transfer request occurs while transfer is enabled (DME = 1, NMIF = 0, DE = 1).
- B. Sets the RV bit corresponding to the mailbox (MBx) to 1.
- C. Suspends data transfer if an NMI interrupt occurs or if the DME bit or DE bit is cleared to 0.

(For details, refer to section 12.3.3, Transfer Suspension and Resumption.) If a transfer request to the high priority DMA channel occurs during a mailbox (MBx) transfer, channel 72 enters transfer wait state.

Channel 74: The number of mailbox (MBx) to be transferred to the transmit wait register is set by software. If the transfer enable state is then entered, the specified mailboxes are transferred according to the mailbox priority (Priority: RCAN\_A-MBx1  $\rightarrow$  RCAN\_A-MBx2  $\rightarrow$ ...).

Data transfer is performed in the following sequence. Figure 12.8 shows the flowchart of data transfer

- 1. Sets a bit in ADMATVPR corresponding to the number of mailbox and sets the DE bit to 1. If the DE bit is set to 1 without specifying mailbox (MBx), performs the operation in step 5 below.
- 2. Performs one mailbox (MBx) transfer corresponding to the TV bit while transfer is enabled (DME = 1, NMIF = 0, DE = 1, TE = 0).
- 3. Clears the TV bit corresponding to the mailbox (MBx) to be transferred to 0.
- 4. Repeats operations in steps 2 and 3 until all the TV bits are cleared to 0.

- 5. Sets the TE bit to 1 if all the TV bits are cleared to 0 and if the specified transfer has been completed. In this case, if an interrupt is enabled (IE = 1), an interrupt is requested to the CPU.
- 6. While the TE bit is set to 1, transfer is not performed even if the TV bit is set.
- 7. Suspends data transfer if an NMI interrupt occurs or if the DME bit or DE bit is cleared to 0.

(For details, refer to section 12.3.3, Transfer Suspension and Resumption.) If a transfer request to the high DMA channel occurs during a mailbox (MBx) transfer, channel 74 enters transfer wait state.

#### Notes

In a receive transfer, data transfer is enabled when the RV bit is cleared. Accordingly, the alias area must be referenced before the RV bit is cleared to avoid conflict.

Note that the order that MBx in which RCAN becomes full does not match the order in which MBx in RCAN is transferred to the alias area. This is because the A-DMAC does not accept a transfer in MBx units if the RV bit is set to 1 and because the algorithm that selects MBx is not a FIFO.

In a transmit transfer, the TE bit is set to 1 without transferring data if transfer is enabled while all the TV bits are cleared to 0. A write to a TV bit is prohibited in the transfer enable state.

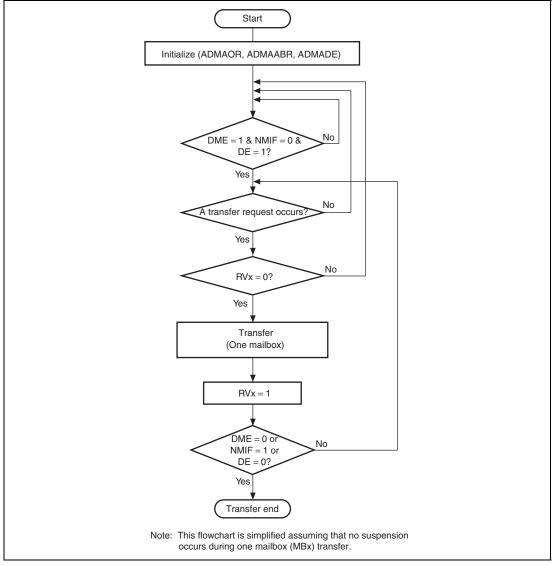


Figure 12.7 DMA Transfer Flowchart (Channel for RCAN Reception)

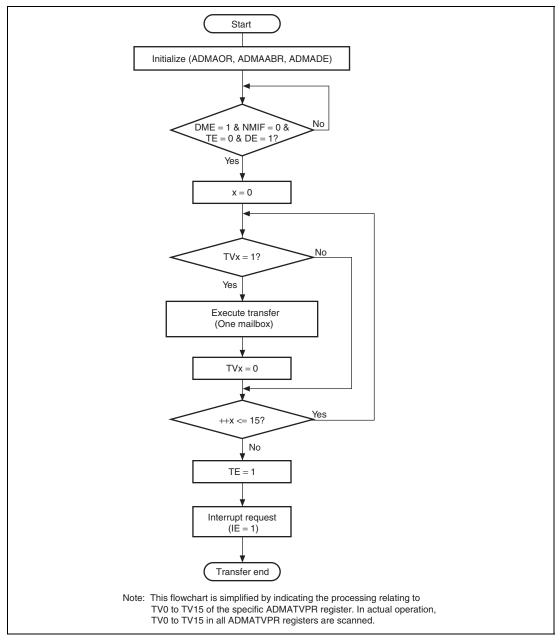


Figure 12.8 DMA Transfer Flowchart (Channel for RCAN Transmission)

#### 12.3.3 Transfer Suspension and Resumption

If one of the sources listed in table 12.13 occurs, data transfer is disabled (transfer is suspended). If it occurs during DMA transfer in a transfer unit, the DMA transfer in a transfer unit is performed correctly and registers such as ADMADV, ADMATE, ADMATCR, ADMAAR, ADMARVPR, and ADMATVPR are also updated correctly.

**Table 12.13 Source of Transfer Suspension** 

| Source  | Channels to be Suspended |
|---|--------------------------|
| NMI interrupt occurrence (NMIF = 1 in the DMAC) | All channels             |
| Clearing the DME bit to 0                       | _                        |
| Clearing the DE bit to 0                        | Corresponding channel    |

If one of the sources listed in table 12.13 is cancelled, A-DMAC enters the transfer enable state (transfer request wait state). A-DMAC channels for the RSPI and SCI refer to the current values for the next transfer request, and perform transfer (transfer is resumed).

If the transfer is suspended during MBx transfer, the A-DMAC channel for RCAN operation differs depending on the source of the transfer suspension, as follows.

An NMI interrupt is cancelled: The next data of MBx being transferred is transferred (transfer is resumed).

The DME (DE) bit is set to 1: Data is transferred from the start of MBx. (In the A-DMA channel for RCAN reception, MBx is informed synchronously with the next transfer request. In the A-DMAC channel for RCAN transmission, MBx has the highest priority.)

## 12.3.4 Bus Operation in Data Transfer by A-DMAC

#### Addressing mode

Performs the operation corresponding to the dual address mode in DMAC. Accordingly, in the read cycle, the transfer source is accessed; while in the write cycle, the transfer destination is accessed. The transfer size is fixed in each channel.

#### • Bus cycle

The A-DMAC requests the bus mastership of I bus if it receives a transfer request. Upon acquiring the I-bus mastership, the A-DMAC starts one unit of DMA transfer consisting of two bus cycles of read and write. In one unit of DMA transfer, the A-DMAC does not release the I-bus mastership. Figure 12.9 shows the data for data transfer performed from the on-chip peripheral module to the on-chip RAM.

Note: In the A-DMAC channels for ATU-III (timers A and F) and the RSPI, one unit of DMA transfer does not include two bus cycles.

#### • Bus mode

The A-DMAC performs data transfer in cycle steal mode. The A-DMAC releases the I-bus mastership after one unit of DMA transfer even if another A-DMAC channel waits for DMA transfer (for details, refer to section 12.3.5, Channel Priorities).

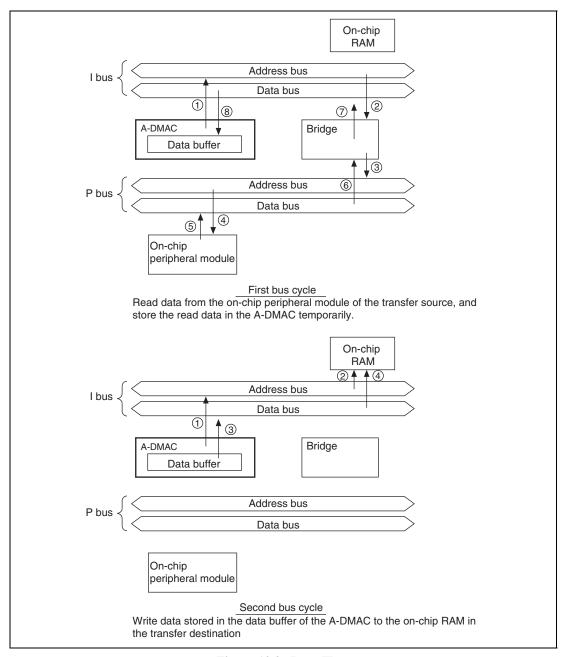


Figure 12.9 Data Flow

## (1) Supplementary Description for A-DMAC Channels Used for ATU-III (timer A)

One unit of DMA transfer consists of three bus cycles of read, write, and write. Figure 12.10 shows the data flow of the A-DMAC channel for timer A.

Note: Because the ADMABUF value, which is transferred to the alias area after DE is changed from 0 to 1, is not the input captured value, the ADMABUF value has no meaning. Accordingly, this register value must be ignored.

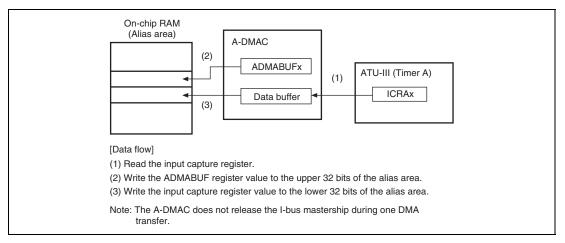


Figure 12.10 Data Flow in Channels for Timer A

# (2) Supplementary Description for A-DMAC Channels Used for ATU-III (timer F)

When the MODE bit in ADMAMODE is set to 1, one unit of DMA transfer consists of four bus cycles of read, write, read, and write. Figure 12.11 shows the data flow of A-DMAC channel for timer F.

When the MODE bit in ADMAMODE is cleared to 0, one unit of DMA transfer consists of two bus cycles of read and write. For details, refer to section 12.3.4, Bus Operation in Data Transfer by A-DMAC.

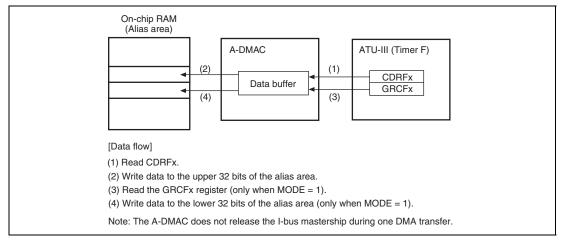


Figure 12.11 Data Flow in Channels for Timer F

#### (3) Supplementary Description for A-DMAC Channels Used for RSPI

Because the RSPI is connected to the A-DMAC via the specific bus, the A-DMAC can access the RSPI data register (SPDR) without the intervention of the P bus. Accordingly, one unit of DMA transfer in the transmission channel can be performed in one bus cycle of read from the I bus (alias area read) and that in the reception channel can be performed in one bus cycle of write to the I bus (alias area write). As a result, high-speed data transfer between the RSPI and on-chip RAM can be performed.

Figure 12.12 shows the data flow of A-DMAC channel for the RSPI.

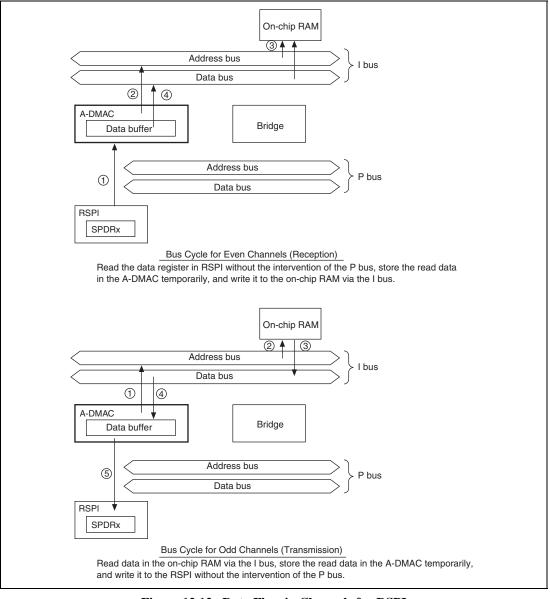


Figure 12.12 Data Flow in Channels for RSPI

#### 12.3.5 Channel Priorities

If multiple transfer requests to multiple channels occur simultaneously, the transfer request is accepted according to the channel priority. The channel priority is fixed (channel 0 > channel 1 >  $\dots$  > channel 74).

If a transfer request for a channel of higher priority occurs during data transfer in the channel for the RCAN, the data transfer in channel for the RCAN is suspended and the data transfer of the high-priority channel is performed. If a transfer request for a RCAN reception channel occurs during data transfer in the RCAN transmission channel, the current data transfer in the RCAN transmission channel is suspended and enters the wait state until data transfer in the RCAN reception channel has been completed.

Figure 12.13 shows the A-DMAC operation when a transfer request for channels 0 and 72 occurs simultaneously and when a transfer request for channel 1 occurs during channel 72 operation.

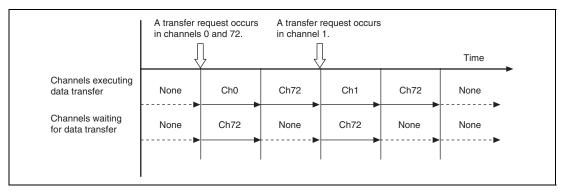


Figure 12.13 Channel Priorities

#### 12.3.6 Transfer Enable/Disable Conditions and Interrupt Requests

Table 12.14 lists the transfer enable and disable conditions for each channel.

The A-DMAC uses an interrupt request signal from the peripheral module as a transfer activation source.

The A-DMAC requests an interrupt under the following two conditions.

#### **(1)** Transfer End Interrupt by the A-DMAC (other than channels 0, 1, and 72)

In the transfer enable state, an interrupt request signal from the peripheral module is used as an A-DMAC transfer activation source and is not informed to the INTC. If the A-DMAC transfer using

the interrupt source has been completed, an A-DMAC transfer end is notified to the INCT (transfer end interrupt) at the timing when the DV or TE bit is set to 1. The transfer end interrupt is cleared by clearing the TE or DV bit to 0.

A transfer end interrupt can be masked by clearing the IE bit to 0.

# (2) Passing through of Transfer Request Interrupts from Peripheral Modules (other than channels 72 and 74)

Interrupt request signals from peripheral modules are masked by the A-DMAC in the 'transfer-disabled state (masked)' shown in table 12.14. On the other hand, interrupt request signals from peripheral modules are directly informed to the INTC in the 'transfer-disabled state (pass-through)' shown in table 12.14. The transfer requests that have been passed through cannot be masked even if the IE bit is cleared to 0.

#### Notes:

- For details on the connections between the A-DMAC and INTC and the transfer request masking by the A-DMAC and DMAC, refer to section 8, Interrupt Controller (INTC).
- The INTC handles interrupt requests from each A-DMAC channel as different interrupt sources.
  - The INTC assigns an A-DMAC transfer end interrupt and transfer request interrupt (pass-through) for the same interrupt from the peripheral module to the same interrupt vector.
- Transfer request interrupts from channels 8, 12, 16, 20 and 24, which correspond to GRCx0 of timer C, are informed to the INTC via the DMAC. Transfer request interrupts from other channels are informed to the INTC directly.
- In the transfer-disabled (masked) state, a DMA transfer request is masked by the A-DMAC. In this case, data transfer is not performed and the transfer request interrupt is not informed to the INTC.
- If the transfer-suspended state is entered (for details, refer to section 12.3.3, Transfer Suspension and Resumption) while the transfer end interrupt is not yet cleared, the interrupt request is retained.

Table 12.14 (1) Transfer Enable and Disable Conditions for Channels Used for ATU-III (Timer G) and ADC

| NMI | DME | TCR      | Transfer Request                 |  |
|-----|-----|----------|----------------------------------|--|
| 1   | ×   | ×        | Transfer disabled (masked)       |  |
| 0   | 0 × |          | Transfer disabled (pass-through) |  |
|     | 1   | TCR = 0  | Transfer disabled (pass-through) |  |
|     |     | TCR != 0 | Transfer enabled                 |  |

[Legend]

x: Don't care

Table 12.14 (2) Transfer Enable and Disable Conditions for Channels Used for ATU-III (Timers A, C, and F)

| NMI | DME | DE | DV | Transfer Request                 |
|-----|-----|----|----|----------------------------------|
| 1   | ×   | ×  | ×  | Transfer disabled (masked)       |
| 0   | 0   | ×  | ×  | Transfer disabled (pass-through) |
|     | 1   | 0  | ×  | Transfer disabled (pass-through) |
|     |     | 1  | ×  | Transfer enabled                 |

[Legend]

x: Don't care

Table 12.14 (3) Transfer Enable and Disable Conditions for Channels Used for RSPI and SCI

| NMI | DME | TCR      | TE | Transfer Request                 |
|-----|-----|----------|----|----------------------------------|
| 1   | ×   | ×        | ×  | Transfer disabled (masked)       |
| 0   | 0   | ×        | ×  | Transfer disabled (pass-through) |
|     | 1   | TCR = 0  | 0  | Transfer disabled (pass-through) |
|     |     |          | 1  | Transfer disabled (masked)       |
|     |     | TCR != 0 | 0  | Transfer enabled                 |
|     |     |          | 1  | Transfer disabled (masked)       |

[Legend]

x: Don't care

Table 12.14 (4) Transfer Enable and Disable Conditions for Channels Used for RCAN (Reception)

| NMI | DME | DE | Transfer Request  |  |
|-----|-----|----|-------------------|--|
| 1   | ×   | ×  | Transfer disabled |  |
| 0   | 0   | ×  | Transfer disabled |  |
|     | 1   | 0  | Transfer disabled |  |
|     |     | 1  | Transfer enabled  |  |

[Legend]

x: Don't care

Table 12.14 (5) Transfer Enable and Disable Conditions for Channels Used for RCAN (Transmission)

| NMI | DME | DE | TE | Transfer Request  |  |
|-----|-----|----|----|-------------------|--|
| 1   | ×   | ×  | ×  | Transfer disabled |  |
| 0   | 0   | ×  | ×  | Transfer disabled |  |
|     | 1   | 0  | ×  | Transfer disabled |  |
|     |     | 1  | 0  | Transfer enabled  |  |
|     |     |    | 1  | Transfer disabled |  |

[Legend]

x: Don't care

# 12.4 Usage Note

In ADMADV, ADMATE, and ADMARVPR, when the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally.

In this case, if 0 (in ADMADV or ADMATE) or 1 (in ADMARVPR) is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU.

To avoid the malfunction, the clear conditions must be observed, described in the notes of ADMADV, ADMATE, and ADMARVPR.

Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

# Section 13 Advanced Timer Unit III (ATU-III)

ATU-III consists of nine timer blocks (timer A to timer J), prescalers, and a controller. The timer blocks have different functions and each can operate independently; timer blocks can also be linked via the clock bus. Each timer block consists of one or more timer subblocks and each subblock has one or more channels.

#### 13.1 Features

- Processing of up to 106 pulse input/output signals
- 153 interrupt sources can be generated. This enables to activate the direct memory access controller (DMAC), automotive direct memory access controller (A-DMAC), and interrupt processing by the CPU.
- 22 pulse output dedicated for A/D (16 for timer D and 6 for timer G)
- On-chip 4-channel prescaler provided, which generates four types of clocks by dividing onchip peripheral clock (Pφ) by 1/1 to 1/1024
- Each channel for a timer can select a count source from among four divided clocks generated by prescaler, two external clocks, and angle clock generated by timer B.

#### (1) Timer A

Timer A has a 32-bit free-run counter and six 32-bit input capture registers. Features are shown below.

- Detection by rising edges, falling edges, or both edges
- A-DMAC activation at capture timing
- Noise canceling function for each external pin with maximum length of 1.64 ms
- Capture interrupt and counter overflow interrupt are available

# (2) Timer B

Timer B consists of three subblocks: an edge-interval measuring block, frequency-multiplied clock generator, and frequency-multiplied clock signal corrector.

1. The edge-interval measuring block is provided with a 32-bit input edge-interval measuring timer, output compare and input capture registers (three registers), 8-bit event counter, and output compare register. This provides the following operations:

- Capture by edges of external event input (rising edge, falling edge, or both edges are selectable)
- Capture by event compare match of external event input
- Capture interrupt and compare match interrupt (edge-interval compare match, event compare match)
- 2. The frequency-multiplied clock generator is provided with 24-bit reloadable counter, reload register, 20-bit multiplied clock counter, and output compare register. This provides the following operations:
- Reloadable counting of values captured by edge-interval measuring block with arbitrary number (1 to 4095)
- Internal clock generated by the underflow of reloadable counter can be used as input of 20-bit multiplied clock counter
- Compare match interrupt generation.
- 3. Frequency-multiplied clock signal corrector is provided with 20-bit correcting event counter, 20-bit correcting multiplied clock counter, multiplied-and-corrected clock generating counter, and correcting counter clearing register. This provides the following operations:
- Frequency-multiplied correcting clock that serves as the count source for other timers can be generated based on the read count in frequency-multiplied clock generator.
- Free-run counter for timer D can be cleared by multiplied-and-corrected clock generating counter and correcting counter clearing register

#### (3) Timer C

Timer C consists of five subblocks that have the same functions. Each subblock consists of four channels. Each subblock is provided with one 24-bit free running counter and four 24-bit general registers. This provides the following operations:

- Input capture or output compare is selectable
- Detection edge for input capture is selectable from among rising edge, falling edge, or both edges.
- Each input capture trigger input has a function that cancels noise with maximum length of 1.64 ms.
- 1, 0, or toggle can be output by a compare match.
- Three PWM waveforms can be output for each subblock in PWM mode.
- Input capture/compare match interrupt and overflow interrupt can be generated. 20 input capture/compare match interrupts activate A-DMAC and five interrupts activate DMAC.

#### Timer D **(4)**

Timer D consists of four subblocks that have the same function. Each subblock consists of four channels. Each subblock is provided with two 24-bit free run counter, offset base register, and four channels. Each channel is provided with four 24-bit output compare registers, four general registers, and four 24-bit down counter for outputting one-shot pulse. This provides the following operations:

- Enables to start downcounter by software. One-shot pulse can also be generated.
- Compare match between compare match register and general register can be used as the start trigger for downcounter. One-shot pulse with offset can also be generated.
- Compare match in general register can stop downcounter and forcibly cut off one-shot pulse output.
- Compare match between compare match register and general register can be output.
- Compare match in compare match register can be used as a trigger and the count number can be captured by general register.
- Free run value can be captured triggered by timer A
- Provided with counter clearing function from timer B
- Generation of sixteen compare match interrpts, eight counter overflow interrupts, and sixteen underflow interrupts. The sixteen undeflow interrupts correspond to DMAC activation.
- Output pulses that show interrupts of compare match A or compare match B for A/D activation. (eight for each)
- Output waveform can be inverted.

#### Timer E (5)

Timer E consists of six subblocks that have the same function. Each subblock consists of four channels. Each channel is provided with 16-bit free running counter, duty cycle setting register, cycle setting register, duty cycle reload register, and cycle reload register. This provides the following operations:

- PWM output with programmable cycle time and duty cycle ranging from 0 to 100%
- Switchable between on-state and off-state duty modes
- Values in duty cycle reload register/cycle reload register can be transferred to duty cycle setting register/cycle setting register at every cycle.
- Writing H'0000 to counter can forcibly end PWM cycle and start new PWM cycle.
- Periodic output of interrupt requests (up to twenty four; of these, six can be used to activate the DMAC.)

#### (6) Timer F

Timer F consists of 20 subblocks. Each subblock is provided with two 24-bit counters, a 16-bit counter, three 24-bit general registers (for subblocks 12 to 15 only, two for other subblocks), and a 16-bit general register. This provides the following operations:

- Noise canceling function for each external pin with maximum length of 1.64 ms
- Seven operation modes: edge counting in a specified period, valid edge interval counting, measurement of time during high/low input levels, measurement of PWM input waveform timing, rotation speed/pulse measurement, up/down event count, and four-time multiplication event count.
- Activates A-DMAC by input capture interrupt
- Overflow interrupt generation

#### (7) Timer G

Timer G consists of six subblocks that have the same function. Each channel is provided with a 16-bit free-run counter and output compare register. This provides the following operations:

- Outputs event that is triggered by compare match. This output can be used as a trigger for AD activation/interrupt.
- Activates DMAC by compare match interrupt

## (8) Timer H

Timer H consists of a pair of 16-bit countr and 16-bit compare match register, and a 32-bit counter. This provides the following operations:

- Measurement of time ranging from 1 to 2<sup>26</sup> times of internal peripheral clock (Pφ) using 16-bit counter and 16-bit compare match register. Can be output as compare match interrupt.
- Equipped with a 32-bit counter to count the compare match occurrence.

# (9) Timer J

Timer J consists of two subblocks that have the same function. Each channel is provided with a 16-bit counter, output compare register, and nine-stage FIFO register. This provides the following operations:

- Detection by rising edge, falling edge or both edges
- Noise canceling function for each external pin with maximum length of 1.64 ms

- Capture the counter value in FIFO register (edge input interval) when detecting edges on external input pin.
- Activates DMAC at the timing when FIFO is full.
- Controls FIFO's effective capture time using compare match register
- FIFI full interrupt, counter overflow, and FIFI overflow interrupt can be generated.

#### 13.2 **Register Addresses**

Addresses of the ATU-III registers are shown below. To access the registers, the following procedure should be followed.

- When writing to reserved bits, the value written must be 0.
- Registers which have more than 16 bits must be read from and written to in 32 bit units. These registers cannot be accessed in 16- or 8-bit units.

**Table 13.1 Common Controller Registers** 

| Address     | 31 24    | 23 16 | 15 8  | 7    | 0 |
|-------------|----------|-------|-------|------|---|
| H'FFFF F000 | ATUENR   |       | CBCNT | NCMR |   |
| H'FFFF F004 | Reserved |       |       |      |   |
| :           |          |       |       |      |   |
| H'FFFF F0FC |          |       |       |      |   |

### **Table 13.2 Prescaler Registers**

| Address     | 31 24    | 23 16 | 15 8  | 7 0 |  |
|-------------|----------|-------|-------|-----|--|
| H'FFFF F100 | PSCR0    |       | PSCR1 |     |  |
| H'FFFF F104 | PS       | CR2   | PSCR3 |     |  |
| H'FFFF F108 | Reserved |       |       |     |  |
| :           |          |       |       |     |  |
| H'FFFF F1FC |          |       |       |     |  |

**Table 13.3** Timer A Registers

| Address     | 31 24    | 23 16 | 15 8   | 7 0      |  |
|-------------|----------|-------|--------|----------|--|
| H'FFFF F200 | Reserved |       | TCRA   | Reserved |  |
| H'FFFF F204 | TIO      | R1A   | TIOI   | R2A      |  |
| H'FFFF F208 | TSRA     | TIERA | Rese   | erved    |  |
| H'FFFF F20C |          | Rese  | rved   |          |  |
| H'FFFF F210 | NCNTA0   | NCRA0 | NCNTA1 | NCRA1    |  |
| H'FFFF F214 | NCNTA2   | NCRA2 | NCNTA3 | NCRA3    |  |
| H'FFFF F218 | NCNTA4   | NCRA4 | NCNTA5 | NCRA5    |  |
| H'FFFF F21C |          | Rese  | rved   |          |  |
| H'FFFF F220 |          | TCNTA |        |          |  |
| H'FFFF F224 |          | Rese  | rved   |          |  |
| H'FFFF F228 |          | ICR   | A0     |          |  |
| H'FFFF F22C |          | ICR   | A1     |          |  |
| H'FFFF F230 |          | ICR   | A2     |          |  |
| H'FFFF F234 |          | ICR   | A3     |          |  |
| H'FFFF F238 |          | ICR   | A4     |          |  |
| H'FFFF F23C | ICRA5    |       |        |          |  |
| H'FFFF F240 | Reserved |       |        |          |  |
| :           |          |       |        |          |  |
| H'FFFF F2FC |          |       |        |          |  |

## **Table 13.4** Timer B Registers

| Address     | 31 24  | 23 16            | 15 8 | 7 0   |
|-------------|--------|------------------|------|-------|
| H'FFFF F300 |        | Rese             | rved |       |
| H'FFFF F304 | TCRB   | TIORB            | TSRB | TIERB |
| H'FFFF F308 |        | Rese             | rved | •     |
| :           |        |                  |      |       |
| H'FFFF F30C |        |                  |      |       |
| H'FFFF F310 |        | TCN <sup>-</sup> | TB0  |       |
| H'FFFF F314 |        | ICR              | B0   |       |
| H'FFFF F318 |        | OCF              | RB0  |       |
| H'FFFF F31C | TCNTB1 | OCRB1            | Res  | erved |
| H'FFFF F320 |        | ICR              | B1   |       |
| H'FFFF F324 |        | ICR              | B2   |       |
| H'FFFF F328 |        | Rese             | rved |       |
| H'FFFF F32C |        |                  |      |       |
| H'FFFF F330 |        | LD               | В    |       |
| H'FFFF F334 |        | RLI              | DВ   |       |
| H'FFFF F338 | PII    | MR               | Res  | erved |
| H'FFFF F33C |        | TCN              | TB2  |       |
| H'FFFF F340 |        | TCN              | TB6  |       |
| H'FFFF F344 |        | OCF              | RB6  |       |
| H'FFFF F348 |        | OCF              | RB7  |       |
| H'FFFF F34C |        | Rese             | rved |       |
| H'FFFF F350 |        | TCN              | TB3  |       |
| H'FFFF F354 |        | TCN <sup>-</sup> | TB4  |       |
| H'FFFF F358 | TCNTB5 |                  |      |       |
| H'FFFF F35C |        | TCCI             | LRB  |       |
| H'FFFF F360 |        | Rese             | rved |       |
| :           |        |                  |      |       |
| H'FFFF F3FC |        |                  |      |       |

**Table 13.5** Timer C Registers

| Address     | 31 24   | 23 16    | 15 8     | 7        | 0 |  |
|-------------|---------|----------|----------|----------|---|--|
| H'FFFF F400 | TSTRC   | Reserved | NCCRC0   | NCCRC1   |   |  |
| H'FFFF F404 | NCCRC2  | NCCRC3   | NCCRC4   | Reserved |   |  |
| H'FFFF F408 |         | Reserved |          |          |   |  |
| H'FFFF F40C | Ī       |          |          |          |   |  |
| H'FFFF F410 | NCNTC00 | NCNTC01  | NCNTC02  | NCNTC03  |   |  |
| H'FFFF F414 | NCRC00  | NCRC01   | NCRC02   | NCRC03   |   |  |
| H'FFFF F418 | NCNTC10 | NCNTC11  | NCNTC12  | NCNTC13  |   |  |
| H'FFFF F41C | NCRC10  | NCRC11   | NCRC12   | NCRC13   |   |  |
| H'FFFF F420 | NCNTC20 | NCNTC21  | NCNTC22  | NCNTC23  |   |  |
| H'FFFF F424 | NCRC20  | NCRC21   | NCRC22   | NCRC23   |   |  |
| H'FFFF F428 | NCNTC30 | NCNTC31  | NCNTC32  | NCNTC33  |   |  |
| H'FFFF F42C | NCRC30  | NCRC31   | NCRC32   | NCRC33   |   |  |
| H'FFFF F430 | NCNTC40 | NCNTC41  | NCNTC42  | NCNTC43  |   |  |
| H'FFFF F434 | NCRC40  | NCRC41   | NCRC42   | NCRC43   |   |  |
| H'FFFF F438 |         | Rese     | erved    |          |   |  |
| H'FFFF F43C |         |          |          |          |   |  |
| H'FFFF F440 | TCRC0   | TIERC0   | TIOF     | RC0      |   |  |
| H'FFFF F444 | TSRC0   |          | Reserved |          |   |  |
| H'FFFF F448 |         | GRO      | C00      |          |   |  |
| H'FFFF F44C |         | GRO      | C01      |          |   |  |
| H'FFFF F450 |         | GRO      | C02      |          |   |  |
| H'FFFF F454 |         | GRO      | C03      |          |   |  |
| H'FFFF F458 |         | TCN      | TC0      |          |   |  |
| H'FFFF F45C |         | Rese     | erved    |          |   |  |
| H'FFFF F460 | TCRC1   | TIERC1   | TIOF     | RC1      |   |  |
| H'FFFF F464 | TSRC1   |          | Reserved |          |   |  |
| H'FFFF F468 |         | GRO      | C10      |          |   |  |
| H'FFFF F46C |         | GRO      | C11      |          |   |  |
| H'FFFF F470 | GRC12   |          |          |          |   |  |
| H'FFFF F474 |         | GR       | C13      |          |   |  |
| H'FFFF F478 |         | TCN      | TC1      |          |   |  |

| Address     | 31 24 | 23 16  | 15      | 8 7    | 0 |
|-------------|-------|--------|---------|--------|---|
| H'FFFF F47C |       | Rese   | rved    | •      |   |
| H'FFFF F480 | TCRC2 | TIERC2 |         | TIORC2 |   |
| H'FFFF F484 | TSRC2 |        | Reserve | ed     |   |
| H'FFFF F488 |       | GR     | C20     |        |   |
| H'FFFF F48C |       | GR     | C21     |        |   |
| H'FFFF F490 |       | GR     | C22     |        |   |
| H'FFFF F494 |       | GR     | C23     |        |   |
| H'FFFF F498 |       | TCN    | TC2     |        |   |
| H'FFFF F49C |       | Rese   | rved    |        |   |
| H'FFFF F4A0 | TCRC3 | TIERC3 |         | TIORC3 |   |
| H'FFFF F4A4 | TSRC3 |        | Reserve | ed     |   |
| H'FFFF F4A8 |       | GR     | C30     |        |   |
| H'FFFF F4AC |       | GRO    | C31     |        |   |
| H'FFFF F4B0 |       | GR     | C32     |        |   |
| H'FFFF F4B4 |       | GRO    | C33     |        |   |
| H'FFFF F4B8 |       | TCN    | TC3     |        |   |
| H'FFFF F4BC |       | Rese   | erved   |        |   |
| H'FFFF F4C0 | TCRC4 | TIERC4 |         | TIORC4 |   |
| H'FFFF F4C4 | TSRC4 |        | Reserve | ed     |   |
| H'FFFF F4C8 |       | GR     | C40     |        |   |
| H'FFFF F4CC |       | GRO    | C41     |        |   |
| H'FFFF F4D0 |       | GRO    | C42     |        |   |
| H'FFFF F4D4 |       | GR     | C43     |        |   |
| H'FFFF F4D8 |       | TCN    | TC4     |        |   |
| H'FFFF F4DC |       | Rese   | rved    |        |   |
| H'FFFF F4E0 |       | Rese   | erved   |        |   |
| :           |       |        |         |        |   |
| H'FFFF F4FC |       |        |         |        |   |

**Table 13.6 Timer D Registers** 

| Address     | 31 24    | 23 16  | 15 8             | 7 0      |
|-------------|----------|--------|------------------|----------|
| H'FFFF F500 | TSTRD    |        | Reserved         |          |
| H'FFFF F504 | Reserved |        |                  |          |
| :           |          |        |                  |          |
| H'FFFF F51F |          |        |                  |          |
| H'FFFF F520 |          | TCNT   | 1D0              |          |
| H'FFFF F524 |          | TCNT   | <sup>-</sup> 2D0 |          |
| H'FFFF F528 |          | OSBI   | RD0              | _        |
| H'FFFF F52C | TCF      | RD0    | TOCRD0           | CMPOD0   |
| H'FFFF F530 |          | TCNT   | 1D1              |          |
| H'FFFF F534 |          | TCNT   | <sup>-</sup> 2D1 |          |
| H'FFFF F538 |          | OSBI   | RD1              |          |
| H'FFFF F53C | TCI      | RD1    | TOCRD1           | CMPOD1   |
| H'FFFF F540 |          | TCNT   | 1D2              |          |
| H'FFFF F544 |          | TCNT   | <sup>-</sup> 2D2 |          |
| H'FFFF F548 |          | OSBI   | RD2              | _        |
| H'FFFF F54C | TCI      | RD2    | TOCRD2           | Reserved |
| H'FFFF F550 |          | TCNT   | 1D3              |          |
| H'FFFF F554 |          | TCNT   | <sup>-</sup> 2D3 |          |
| H'FFFF F558 |          | OSBI   | RD3              |          |
| H'FFFF F55C | TCI      | RD3    | TOCRD3           | Reserved |
| H'FFFF F560 |          | Rese   | rved             |          |
| :           |          |        |                  |          |
| H'FFFF F57C |          |        | 1                |          |
| H'FFFF F580 | TIOF     | R1D0   | TIOF             | R2D0     |
| H'FFFF F584 | Reserved | DSTRD0 | Reserved         | DSRD0    |
| H'FFFF F588 | DCRD0    |        | Reserved         |          |
| H'FFFF F58C | TSI      | RD0    |                  | RD0      |
| H'FFFF F590 | OCRD00   |        |                  |          |
| H'FFFF F594 | OCRD01   |        |                  |          |
| H'FFFF F598 |          | OCR    | D02              |          |

| Address     | 31 24    | 1 23 16          | 15 8     | 7 0   |  |  |
|-------------|----------|------------------|----------|-------|--|--|
| H'FFFF F59C |          | OCR              | D03      | •     |  |  |
| H'FFFF F5A0 |          | GRD00            |          |       |  |  |
| H'FFFF F5A4 |          | GRD01            |          |       |  |  |
| H'FFFF F5A8 |          | GRD02            |          |       |  |  |
| H'FFFF F5AC |          | GRI              | 003      |       |  |  |
| H'FFFF F5B0 |          | DCN              | ΓD00     |       |  |  |
| H'FFFF F5B4 |          | DCN              | ΓD01     |       |  |  |
| H'FFFF F5B8 |          | DCN              | ΓD02     |       |  |  |
| H'FFFF F5BC |          | DCN <sup>-</sup> | TD03     |       |  |  |
| H'FFFF F5C0 | TIC      | R1D1             | TIO      | R2D1  |  |  |
| H'FFFF F5C4 | Reserved | DSTRD1           | Reserved | DSRD1 |  |  |
| H'FFFF F5C8 | DC       | CRD1             | Res      | erved |  |  |
| H'FFFF F5CC | TS       | SRD1             | TIE      | RD1   |  |  |
| H'FFFF F5D0 |          | OCR              | D10      |       |  |  |
| H'FFFF F5D4 |          | OCR              | D11      |       |  |  |
| H'FFFF F5D8 |          | OCR              | D12      |       |  |  |
| H'FFFF F5DC |          | OCR              | D13      |       |  |  |
| H'FFFF F5E0 |          | GRI              | 010      |       |  |  |
| H'FFFF F5E4 |          | GRI              | 011      |       |  |  |
| H'FFFF F5E8 |          | GRI              | 012      |       |  |  |
| H'FFFF F5EC |          | GRI              | 013      |       |  |  |
| H'FFFF F5F0 |          | DCN              | ΓD10     |       |  |  |
| H'FFFF F5F4 |          | DCN              | ΓD11     |       |  |  |
| H'FFFF F5F8 |          | DCN              | TD12     |       |  |  |
| H'FFFF F5FC |          | DCN              | ΓD13     |       |  |  |
| H'FFFF F600 | TIC      | R1D2             | TIO      | R2D2  |  |  |
| H'FFFF F604 | Reserved | DSTRD2           | Reserved | DSRD2 |  |  |
| H'FFFF F608 | DC       | CRD2             | Res      | erved |  |  |
| H'FFFF F60C | TSRD2    |                  | TIE      | RD2   |  |  |
| H'FFFF F610 |          | OCRD20           |          |       |  |  |
| H'FFFF F614 |          | OCR              | D21      |       |  |  |
| H'FFFF F618 |          | OCR              | D22      | -     |  |  |

| Address     | 31 24    | 23 16    | 15 8     | 7 0   |  |
|-------------|----------|----------|----------|-------|--|
| H'FFFF F61C |          | OCRD23   |          |       |  |
| H'FFFF F620 |          | GRD20    |          |       |  |
| H'FFFF F624 |          | GRI      | D21      |       |  |
| H'FFFF F628 |          | GRI      | 022      |       |  |
| H'FFFF F62C |          | GRI      | 023      |       |  |
| H'FFFF F630 |          | DCN      | ΓD20     |       |  |
| H'FFFF F634 |          | DCN      | ΓD21     |       |  |
| H'FFFF F638 |          | DCN      | ΓD22     |       |  |
| H'FFFF F63C |          | DCN      | ΓD23     |       |  |
| H'FFFF F640 | TIO      | R1D3     | TIOI     | R2D3  |  |
| H'FFFF F644 | Reserved | DSTRD3   | Reserved | DSRD3 |  |
| H'FFFF F648 | DC       | RD3      | Rese     | erved |  |
| H'FFFF F64C | TS       | RD3      | TIE      | RD3   |  |
| H'FFFF F650 |          | OCR      | D30      |       |  |
| H'FFFF F654 |          | OCR      | D31      |       |  |
| H'FFFF F658 |          | OCR      | D32      |       |  |
| H'FFFF F65C |          | OCR      | D33      |       |  |
| H'FFFF F660 |          | GRI      | D30      |       |  |
| H'FFFF F664 |          | GRI      | D31      |       |  |
| H'FFFF F668 |          | GRI      | 032      |       |  |
| H'FFFF F66C |          | GRI      | D33      |       |  |
| H'FFFF F670 |          | DCN      | TD30     |       |  |
| H'FFFF F674 |          | DCN      | TD31     |       |  |
| H'FFFF F678 | DCNTD32  |          |          |       |  |
| H'FFFF F67C | DCNTD33  |          |          |       |  |
| H'FFFF F680 |          | Reserved |          |       |  |
| :           |          |          |          |       |  |
| H'FFFF F6FF |          |          |          |       |  |

**Table 13.7 Timer E Registers** 

| Address     | 31 24    | 23 16        | 15 8     | 7 0     |  |
|-------------|----------|--------------|----------|---------|--|
| H'FFFF F700 | TSTRE    |              | Reserved |         |  |
| H'FFFF F704 | Reserved |              |          |         |  |
| :           |          |              |          |         |  |
| H'FFFF F7FC |          |              |          |         |  |
| H'FFFF F800 | TCRE0    | TOCRE0       | TIERE0   | RLDCRE0 |  |
| H'FFFF F804 | TSRE0    |              | Reserved |         |  |
| H'FFFF F808 | PSCRE0   |              | Reserved |         |  |
| H'FFFF F80C | SSTRE0   |              | Reserved |         |  |
| H'FFFF F810 | CYLE     | RE00         | CYL      | RE01    |  |
| H'FFFF F814 | CYLF     | RE02         | CYL      | RE03    |  |
| H'FFFF F818 | DTF      | RE00         | DTF      | RE01    |  |
| H'FFFF F81C | DTF      | RE02         | DTF      | RE03    |  |
| H'FFFF F820 | CRLI     | DE00 CRLDE01 |          | DE01    |  |
| H'FFFF F824 | CRLI     | DE02         | CRLDE03  |         |  |
| H'FFFF F828 | DRLI     | DE00         | DRLDE01  |         |  |
| H'FFFF F82C | DRLI     | DE02         | DRLDE03  |         |  |
| H'FFFF F830 | TCN      | TE00         | TCNTE01  |         |  |
| H'FFFF F834 | TCN      | TE02         | TCN      | TCNTE03 |  |
| H'FFFF F838 |          | Rese         | rved     |         |  |
| H'FFFF F83C |          |              |          |         |  |
| H'FFFF F840 | TCRE1    | TOCRE1       | TIERE1   | RLDCRE1 |  |
| H'FFFF F844 | TSRE1    |              | Reserved |         |  |
| H'FFFF F848 | PSCRE1   |              | Reserved |         |  |
| H'FFFF F84C | SSTRE1   |              | Reserved |         |  |
| H'FFFF F850 | CYLI     | RE10         | CYL      | RE11    |  |
| H'FFFF F854 | CYLI     | RE12         | CYL      | RE13    |  |
| H'FFFF F858 | DTP      | RE10         | DTF      | RE11    |  |
| H'FFFF F85C | DTP      | RE12         | DTRE13   |         |  |
| H'FFFF F860 | CRLI     | DE10         | CRL      | DE11    |  |
| H'FFFF F864 | CRLI     | DE12         | CRL      | DE13    |  |

| Address     | 31 24   | 23 16  | 15 8     | 7 0     |
|-------------|---------|--------|----------|---------|
| H'FFFF F868 | DRLI    | DE10   | DRL      | DE11    |
| H'FFFF F86C | DRLI    | DE12   | DRLI     | DE13    |
| H'FFFF F870 | TCN     | TE10   | TCN      | ΓE11    |
| H'FFFF F874 | TCN     | TE12   | TCN      | TE13    |
| H'FFFF F878 |         | Rese   | rved     |         |
| H'FFFF F87C |         |        |          |         |
| H'FFFF F880 | TCRE2   | TOCRE2 | TIERE2   | RLDCRE2 |
| H'FFFF F884 | TSRE2   |        | Reserved |         |
| H'FFFF F888 | PSCRE2  |        | Reserved |         |
| H'FFFF F88C | SSTRE2  |        | Reserved |         |
| H'FFFF F890 | CYLF    | RE20   | CYLF     | RE21    |
| H'FFFF F894 | CYLF    | RE22   | CYLF     | RE23    |
| H'FFFF F898 | DTRE20  |        | DTR      | E21     |
| H'FFFF F89C | DTR     | E22    | DTRE23   |         |
| H'FFFF F8A0 | CRLI    | DE20   | CRLDE21  |         |
| H'FFFF F8A4 | CRLDE22 |        | CRLI     | DE23    |
| H'FFFF F8A8 | DRLI    | DE20   | DRLDE21  |         |
| H'FFFF F8AC | DRLI    | DE22   | DRLDE23  |         |
| H'FFFF F8B0 | TCN     | TE20   | TCNTE21  |         |
| H'FFFF F8B4 | TCN     | TE22   | TCNTE23  |         |
| H'FFFF F8B8 |         | Rese   | rved     |         |
| H'FFFF F8BC |         |        |          |         |
| H'FFFF F8C0 | TCRE3   | TOCRE3 | TIERE3   | RLDCRE3 |
| H'FFFF F8C4 | TSRE3   |        | Reserved |         |
| H'FFFF F8C8 | PSCRE3  |        | Reserved |         |
| H'FFFF F8CC | SSTRE3  |        | Reserved |         |
| H'FFFF F8D0 | CYLF    | RE30   | CYLRE31  |         |
| H'FFFF F8D4 | CYLRE32 |        | CYLRE33  |         |
| H'FFFF F8D8 | DTRE30  |        | DTRE31   |         |
| H'FFFF F8DC | DTRE32  |        | DTRE33   |         |
| H'FFFF F8E0 | CRLI    | DE30   | CRLI     | DE31    |
| H'FFFF F8E4 | CRLI    | DE32   | CRLI     | DE33    |

| Address     | 31 24   | 23 16  | 15 8     | 7 0     | ) |
|-------------|---------|--------|----------|---------|---|
| H'FFFF F8E8 | DRLI    | DE30   | DRLDE31  |         |   |
| H'FFFF F8EC | DRLI    | DE32   | DRLI     | DE33    |   |
| H'FFFF F8F0 | TCN     | TE30   | TCN      | TE31    |   |
| H'FFFF F8F4 | TCN'    | TE32   | TCN      | TE33    |   |
| H'FFFF F8F8 |         | Rese   | rved     |         |   |
| H'FFFF F8FC |         |        |          |         |   |
| H'FFFF F900 | TCRE4   | TOCRE4 | TIERE4   | RLDCRE4 |   |
| H'FFFF F904 | TSRE4   |        | Reserved |         |   |
| H'FFFF F908 | PSCRE4  |        | Reserved |         |   |
| H'FFFF F90C | SSTRE4  |        | Reserved |         |   |
| H'FFFF F910 | CYLI    | RE40   | CYLRE41  |         |   |
| H'FFFF F914 | CYLI    | RE42   | CYLRE43  |         |   |
| H'FFFF F918 | DTF     | RE40   | DTRE41   |         |   |
| H'FFFF F91C | DTF     | RE42   | DTRE43   |         |   |
| H'FFFF F920 | CRLI    | DE40   | CRLDE41  |         |   |
| H'FFFF F924 | CRLI    | DE42   | CRLDE43  |         |   |
| H'FFFF F928 | DRLI    | DE40   | DRLDE41  |         |   |
| H'FFFF F92C | DRLI    | DE42   | DRLDE43  |         |   |
| H'FFFF F930 | TCNTE40 |        | TCNTE41  |         |   |
| H'FFFF F934 | TCNTE42 |        | TCNTE43  |         |   |
| H'FFFF F938 |         | Rese   | rved     |         |   |
| H'FFFF F93C |         |        |          |         |   |

| Address     | 31 24    | 23 16  | 15 8     | 7 0     |  |
|-------------|----------|--------|----------|---------|--|
| H'FFFF F940 | TCRE5    | TOCRE5 | TIERE5   | RLDCRE5 |  |
| H'FFFF F944 | TSRE5    |        | Reserved |         |  |
| H'FFFF F948 | PSCRE5   |        | Reserved |         |  |
| H'FFFF F94C | SSTRE5   |        | Reserved |         |  |
| H'FFFF F950 | CYLI     | RE50   | CYLF     | RE51    |  |
| H'FFFF F954 | CYLI     | RE52   | CYLF     | RE53    |  |
| H'FFFF F958 | DTF      | RE50   | DTRE51   |         |  |
| H'FFFF F95C | DTF      | RE52   | DTRE53   |         |  |
| H'FFFF F960 | CRL      | DE50   | CRLDE51  |         |  |
| H'FFFF F964 | CRLI     | DE52   | CRLDE53  |         |  |
| H'FFFF F968 | DRLI     | DE50   | DRLI     | DE51    |  |
| H'FFFF F96C | DRL      | DE52   | DRLI     | DE53    |  |
| H'FFFF F970 | TCN      | TE50   | TCN      | TE51    |  |
| H'FFFF F974 | TCN      | TE52   | TCN      | TE53    |  |
| H'FFFF F978 | Reserved |        |          |         |  |
| :           |          |        |          |         |  |
| H'FFFF F9FC |          |        |          |         |  |

**Table 13.8 Timer F Registers** 

| Address     | 31 24         | 23 16   | 15 8     | 7 0     |  |
|-------------|---------------|---------|----------|---------|--|
| H'FFFF FA00 |               | TST     | RF       |         |  |
| H'FFFF FA04 |               | NCC     | CRF      |         |  |
| H'FFFF FA08 |               | Rese    | rved     |         |  |
| H'FFFF FA0C | 1             |         |          |         |  |
| H'FFFF FA10 | NCNTFA0       | NCRFA0  | NCNTFA1  | NCRFA1  |  |
| H'FFFF FA14 | NCNTFA2       | NCRFA2  | NCNTFA3  | NCRFA3  |  |
| H'FFFF FA18 | NCNTFA4       | NCRFA4  | NCNTFA5  | NCRFA5  |  |
| H'FFFF FA1C | NCNTFA6       | NCRFA6  | NCNTFA7  | NCRFA7  |  |
| H'FFFF FA20 | NCNTFA8       | NCRFA8  | NCNTFA9  | NCRFA9  |  |
| H'FFFF FA24 | NCNTFA10      | NCRFA10 | NCNTFA11 | NCRFA11 |  |
| H'FFFF FA28 | NCNTFA12      | NCRFA12 | NCNTFA13 | NCRFA13 |  |
| H'FFFF FA2C | NCNTFA14      | NCRFA14 | NCNTFA15 | NCRFA15 |  |
| H'FFFF FA30 | NCNTFA16      | NCRFA16 | NCNTFA17 | NCRFA17 |  |
| H'FFFF FA34 | NCNTFA18      | NCRFA18 | NCNTFA19 | NCRFA19 |  |
| H'FFFF FA38 |               | Rese    | rved     |         |  |
| :           |               |         |          |         |  |
| H'FFFF FA4C |               |         |          |         |  |
| H'FFFF FA50 | NCNTFB0       | NCRFB0  | NCNTFB1  | NCRFB1  |  |
| H'FFFF FA54 | NCNTFB2       | NCRFB2  | Rese     | erved   |  |
| H'FFFF FA58 |               | Rese    | rved     |         |  |
| :           |               |         |          |         |  |
| H'FFFF FA7C |               |         |          |         |  |
| H'FFFF FA80 | TCRF0         | TIERF0  | Reserved | TSRF0   |  |
| H'FFFF FA84 |               | ECN1    | ΓAF0     |         |  |
| H'FFFF FA88 | ECNTBF0 GRBF0 |         |          |         |  |
| H'FFFF FA8C | ECNTCF0       |         |          |         |  |
| H'FFFF FA90 | GRAF0         |         |          |         |  |
| H'FFFF FA94 | CDRF0         |         |          |         |  |
| H'FFFF FA98 |               | GRO     | CF0      |         |  |
| H'FFFF FA9C |               | Rese    | rved     |         |  |

| Address     | 31 24            | 23 16  | 15 8     | 7 0   |  |
|-------------|------------------|--------|----------|-------|--|
| H'FFFF FAA0 | TCRF1            | TIERF1 | Reserved | TSRF1 |  |
| H'FFFF FAA4 |                  | ECNT   | AF1      |       |  |
| H'FFFF FAA8 | ECN <sup>-</sup> | TBF1   | GR       | BF1   |  |
| H'FFFF FAAC |                  | ECNT   | CF1      |       |  |
| H'FFFF FAB0 |                  | GRA    | AF1      |       |  |
| H'FFFF FAB4 |                  | CDF    | RF1      |       |  |
| H'FFFF FAB8 |                  | GRO    | CF1      |       |  |
| H'FFFF FABC |                  | Rese   | rved     |       |  |
| H'FFFF FAC0 | TCRF2            | TIERF2 | Reserved | TSRF2 |  |
| H'FFFF FAC4 |                  | ECNT   | AF2      |       |  |
| H'FFFF FAC8 | ECN <sup>-</sup> | TBF2   | GR       | BF2   |  |
| H'FFFF FACC |                  | ECNT   | CF2      |       |  |
| H'FFFF FAD0 |                  | GRA    | NF2      |       |  |
| H'FFFF FAD4 |                  | CDF    | RF2      |       |  |
| H'FFFF FAD8 |                  | GRO    | F2       |       |  |
| H'FFFF FADC |                  | Rese   | rved     |       |  |
| H'FFFF FAE0 | TCRF3            | TIERF3 | Reserved | TSRF3 |  |
| H'FFFF FAE4 |                  | ECNT   | AF3      |       |  |
| H'FFFF FAE8 | ECN <sup>*</sup> | TBF3   | GR       | BF3   |  |
| H'FFFF FAEC |                  | ECNT   | CF3      |       |  |
| H'FFFF FAF0 |                  | GRA    | AF3      |       |  |
| H'FFFF FAF4 |                  | CDF    | RF3      |       |  |
| H'FFFF FAF8 |                  | GRO    | F3       |       |  |
| H'FFFF FAFC |                  | Rese   | rved     |       |  |
| H'FFFF FB00 | TCRF4            | TIERF4 | Reserved | TSRF4 |  |
| H'FFFF FB04 | ECNTAF4          |        |          |       |  |
| H'FFFF FB08 | ECNTBF4 GRBF4    |        |          |       |  |
| H'FFFF FB0C | ECNTCF4          |        |          |       |  |
| H'FFFF FB10 | GRAF4            |        |          |       |  |
| H'FFFF FB14 | CDRF4            |        |          |       |  |
| H'FFFF FB18 |                  | GRO    | F4       |       |  |
| H'FFFF FB1C |                  | Rese   | rved     |       |  |

| Address     | 31 24            | 23 16  | 15 8     | 7 0   |  |
|-------------|------------------|--------|----------|-------|--|
| H'FFFF FB20 | TCRF5            | TIERF5 | Reserved | TSRF5 |  |
| H'FFFF FB24 |                  | ECNT   | AF5      |       |  |
| H'FFFF FB28 | ECN              | TBF5   | GR       | BF5   |  |
| H'FFFF FB2C |                  | ECNT   | CF5      |       |  |
| H'FFFF FB30 |                  | GRA    | NF5      |       |  |
| H'FFFF FB34 |                  | CDF    | RF5      |       |  |
| H'FFFF FB38 |                  | GRO    | CF5      |       |  |
| H'FFFF FB3C |                  | Rese   | rved     |       |  |
| H'FFFF FB40 | TCRF6            | TIERF6 | Reserved | TSRF6 |  |
| H'FFFF FB44 |                  | ECNT   | AF6      |       |  |
| H'FFFF FB48 | ECN <sup>-</sup> | TBF6   | GR       | BF6   |  |
| H'FFFF FB4C |                  | ECNT   | CF6      |       |  |
| H'FFFF FB50 |                  | GRA    | AF6      |       |  |
| H'FFFF FB54 |                  | CDF    | RF6      |       |  |
| H'FFFF FB58 |                  | GRO    | CF6      |       |  |
| H'FFFF FB5C |                  | Rese   | rved     |       |  |
| H'FFFF FB60 | TCRF7            | TIERF7 | Reserved | TSRF7 |  |
| H'FFFF FB64 |                  | ECNT   | AF7      |       |  |
| H'FFFF FB68 | ECN <sup>-</sup> | TBF7   | GR       | BF7   |  |
| H'FFFF FB6C |                  | ECNT   | CF7      |       |  |
| H'FFFF FB70 |                  | GRA    | AF7      |       |  |
| H'FFFF FB74 |                  | CDF    | RF7      |       |  |
| H'FFFF FB78 |                  | GRO    | CF7      |       |  |
| H'FFFF FB7C |                  | Rese   | rved     |       |  |
| H'FFFF FB80 | TCRF8            | TIERF8 | Reserved | TSRF8 |  |
| H'FFFF FB84 | ECNTAF8          |        |          |       |  |
| H'FFFF FB88 | ECNTBF8 GRBF8    |        |          |       |  |
| H'FFFF FB8C | ECNTCF8          |        |          |       |  |
| H'FFFF FB90 | GRAF8            |        |          |       |  |
| H'FFFF FB94 | CDRF8            |        |          |       |  |
| H'FFFF FB98 |                  | GRO    | CF8      |       |  |
| H'FFFF FB9C |                  | Rese   | rved     |       |  |

| Address     | 31 24    | 23 16   | 15 8     | 7 0    |  |
|-------------|----------|---------|----------|--------|--|
| H'FFFF FBA0 | TCRF9    | TIERF9  | Reserved | TSRF9  |  |
| H'FFFF FBA4 |          | ECNT    | AF9      |        |  |
| H'FFFF FBA8 | ECN      | TBF9    | GRI      | BF9    |  |
| H'FFFF FBAC |          | ECNT    | CF9      |        |  |
| H'FFFF FBB0 |          | GRA     | NF9      |        |  |
| H'FFFF FBB4 |          | CDF     | RF9      |        |  |
| H'FFFF FBB8 |          | GRO     | CF9      |        |  |
| H'FFFF FBBC |          | Rese    | rved     |        |  |
| H'FFFF FBC0 | TCRF10   | TIERF10 | Reserved | TSRF10 |  |
| H'FFFF FBC4 |          | ECNT    | AF10     |        |  |
| H'FFFF FBC8 | ECNT     | BF10    | GRE      | GRBF10 |  |
| H'FFFF FBCC |          | ECNT    | CF10     |        |  |
| H'FFFF FBD0 |          | GRA     | F10      |        |  |
| H'FFFF FBD4 |          | CDR     | F10      |        |  |
| H'FFFF FBD8 |          | GRC     | F10      |        |  |
| H'FFFF FBDC |          | Rese    | rved     |        |  |
| H'FFFF FBE0 | TCRF11   | TIERF11 | Reserved | TSRF11 |  |
| H'FFFF FBE4 |          | ECNT    | AF11     |        |  |
| H'FFFF FBE8 | ECNT     | BF11    | GRE      | BF11   |  |
| H'FFFF FBEC | ECNTCF11 |         |          |        |  |
| H'FFFF FBF0 | GRAF11   |         |          |        |  |
| H'FFFF FBF4 | CDRF11   |         |          |        |  |
| H'FFFF FBF8 | GRCF11   |         |          |        |  |
| H'FFFF FBFC |          | Rese    | rved     |        |  |

| Address     | 31 24                    | 23 16        | 15 8          | 7      | 0 |
|-------------|--------------------------|--------------|---------------|--------|---|
| H'FFFF FC00 | TCRF12                   | TIERF12      | Reserved      | TSRF12 |   |
| H'FFFF FC04 |                          | ECNT         | AF12          |        |   |
| H'FFFF FC08 | ECNT                     | BF12         | GRE           | 3F12   |   |
| H'FFFF FC0C |                          | ECNT         | CF12          |        |   |
| H'FFFF FC10 |                          | GRA          | F12           |        |   |
| H'FFFF FC14 |                          | CDR          | F12           |        |   |
| H'FFFF FC18 |                          | GRC          | F12           |        |   |
| H'FFFF FC1C |                          | GRDF12 (only | / F12 to F15) |        |   |
| H'FFFF FC20 | TCRF13                   | TIERF13      | Reserved      | TSRF13 |   |
| H'FFFF FC24 |                          | ECNT         | AF13          |        |   |
| H'FFFF FC28 | ECNT                     | BF13         | GRE           | 3F13   |   |
| H'FFFF FC2C |                          | ECNT         | CF13          |        |   |
| H'FFFF FC30 |                          | GRA          | F13           |        |   |
| H'FFFF FC34 |                          | CDR          | F13           |        |   |
| H'FFFF FC38 |                          | GRC          | F13           |        |   |
| H'FFFF FC3C |                          | GRDF13 (only | / F12 to F15) |        |   |
| H'FFFF FC40 | TCRF14                   | TIERF14      | Reserved      | TSRF14 |   |
| H'FFFF FC44 |                          | ECNT         | AF14          |        |   |
| H'FFFF FC48 | ECNT                     | BF14         | GRE           | 3F14   |   |
| H'FFFF FC4C |                          | ECNT         | CF14          |        |   |
| H'FFFF FC50 |                          | GRA          | F14           |        |   |
| H'FFFF FC54 |                          | CDR          | F14           |        |   |
| H'FFFF FC58 |                          | GRC          | F14           |        |   |
| H'FFFF FC5C | GRDF14 (only F12 to F15) |              |               |        |   |
| H'FFFF FC60 | TCRF15                   | TIERF15      | Reserved      | TSRF15 |   |
| H'FFFF FC64 | ECNTAF15                 |              |               |        |   |
| H'FFFF FC68 | ECNTBF15 GRBF15          |              |               |        |   |
| H'FFFF FC6C | ECNTCF15                 |              |               |        |   |
| H'FFFF FC70 | GRAF15                   |              |               |        |   |
| H'FFFF FC74 | CDRF15                   |              |               |        |   |
| H'FFFF FC78 |                          | GRC          | F15           |        |   |
| H'FFFF FC7C |                          | GRDF15 (only | / F12 to F15) |        |   |

| Address     | 31 24           | 23 16   | 15 8     | 7 0    |  |  |
|-------------|-----------------|---------|----------|--------|--|--|
| H'FFFF FC80 | TCRF16          | TIERF16 | Reserved | TSRF16 |  |  |
| H'FFFF FC84 |                 | ECNT    | AF16     |        |  |  |
| H'FFFF FC88 | ECNT            | TBF16   | GRE      | 3F16   |  |  |
| H'FFFF FC8C |                 | ECNT    | CF16     |        |  |  |
| H'FFFF FC90 |                 | GRA     | F16      |        |  |  |
| H'FFFF FC94 |                 | CDR     | F16      |        |  |  |
| H'FFFF FC98 |                 | GRC     | F16      |        |  |  |
| H'FFFF FC9C |                 | Rese    | rved     |        |  |  |
| H'FFFF FCA0 | TCRF17          | TIERF17 | Reserved | TSRF17 |  |  |
| H'FFFF FCA4 |                 | ECNT    | AF17     |        |  |  |
| H'FFFF FCA8 | ECN1            | TBF17   | GRE      | BF17   |  |  |
| H'FFFF FCAC |                 | ECNT    | CF17     |        |  |  |
| H'FFFF FCB0 |                 | GRA     | F17      |        |  |  |
| H'FFFF FCB4 |                 | CDR     | F17      |        |  |  |
| H'FFFF FCB8 |                 | GRC     | F17      |        |  |  |
| H'FFFF FCBC |                 | Rese    | rved     |        |  |  |
| H'FFFF FCC0 | TCRF18          | TIERF18 | Reserved | TSRF18 |  |  |
| H'FFFF FCC4 |                 | ECNT    | AF18     |        |  |  |
| H'FFFF FCC8 | ECN1            | TBF18   | GRE      | 3F18   |  |  |
| H'FFFF FCCC |                 | ECNT    | CF18     |        |  |  |
| H'FFFF FCD0 |                 | GRA     | F18      |        |  |  |
| H'FFFF FCD4 |                 | CDR     | F18      |        |  |  |
| H'FFFF FCD8 |                 | GRC     | F18      |        |  |  |
| H'FFFF FCDC |                 | Rese    | rved     |        |  |  |
| H'FFFF FCE0 | TCRF19          | TIERF19 | Reserved | TSRF19 |  |  |
| H'FFFF FCE4 | ECNTAF19        |         |          |        |  |  |
| H'FFFF FCE8 | ECNTBF19 GRBF19 |         |          |        |  |  |
| H'FFFF FCEC | ECNTCF19        |         |          |        |  |  |
| H'FFFF FCF0 | GRAF19          |         |          |        |  |  |
| H'FFFF FCF4 | CDRF19          |         |          |        |  |  |
| H'FFFF FCF8 |                 | GRC     | F19      |        |  |  |
| H'FFFF FCFC |                 | Rese    | rved     |        |  |  |

| Address     | 31 | 24 | 23 | 16   | 15   | 8 | 7 | 0 |
|-------------|----|----|----|------|------|---|---|---|
| H'FFFF FD00 |    |    |    | Rese | rved |   |   |   |
| :           |    |    |    |      |      |   |   |   |
| H'FFFF FDFC |    |    |    |      |      |   |   |   |

Table 13.9 Timer G Registers

| Address     | 31 24                   | 23 16 | 15 8  | 7 0   |
|-------------|-------------------------|-------|-------|-------|
| H'FFFF FE00 | Reserved TSTRG Reserved |       |       |       |
| H'FFFF FE04 |                         | Rese  | rved  |       |
| :           |                         |       |       |       |
| H'FFFF FE7C |                         |       |       |       |
| H'FFFF FE80 | TCRG0                   | TSRG0 | Res   | erved |
| H'FFFF FE84 | TCN                     | ITG0  | 00    | RG0   |
| H'FFFF FE88 |                         | Rese  | rved  |       |
| H'FFFF FE8C |                         |       |       |       |
| H'FFFF FE90 | TCRG1                   | TSRG1 | Res   | erved |
| H'FFFF FE94 | TCN                     | TG1   | OC    | RG1   |
| H'FFFF FE98 |                         | Rese  | rved  |       |
| H'FFFF FE9C |                         |       |       |       |
| H'FFFF FEA0 | TCRG2                   | TSRG2 | Res   | erved |
| H'FFFF FEA4 | TCN                     | TG2   | OC    | RG2   |
| H'FFFF FEA8 |                         | Rese  | rved  |       |
| H'FFFF FEAC |                         |       |       |       |
| H'FFFF FEB0 | TCRG3                   | TSRG3 | Res   | erved |
| H'FFFF FEB4 | TCN                     | TG3   | OCRG3 |       |
| H'FFFF FEB8 |                         | Rese  | rved  |       |
| H'FFFF FEBC |                         |       |       |       |
| H'FFFF FEC0 | TCRG4                   | TSRG4 | Res   | erved |
| H'FFFF FEC4 | TCN                     | ITG4  | OC    | RG4   |
| H'FFFF FEC8 |                         | Rese  | rved  |       |
| H'FFFF FECC |                         |       |       |       |
| H'FFFF FED0 | TCRG5                   | TSRG5 | Res   | erved |
| H'FFFF FED4 | TCN                     | TG5   | OC    | RG5   |
| H'FFFF FED8 |                         | Rese  | rved  |       |
| :           |                         |       |       |       |
| H'FFFF FEFC |                         |       |       |       |

## **Table 13.10 Timer H Registers**

| Address     | 31 24    | 23 16 | 15       | 8 | 7 | 0 |
|-------------|----------|-------|----------|---|---|---|
| H'FFFF FF00 |          | Rese  | rved     |   |   |   |
| :           |          |       |          |   |   |   |
| H'FFFF FF3C |          |       |          |   |   |   |
| H'FFFF FF40 | TCRH     | TSRH  | Reserved |   |   |   |
| H'FFFF FF44 | TC       | NT1H  | OCR1H    |   |   |   |
| H'FFFF FF48 |          | TCN   | T2H      |   |   |   |
| H'FFFF FF4C | Reserved |       |          |   |   |   |
| :           |          |       |          |   |   |   |
| H'FFFF FF7C |          |       |          |   |   |   |

## **Table 13.11 Timer J Registers**

| Address     | 31 24    | 23 16  | 15 8     | 7 0      |  |
|-------------|----------|--------|----------|----------|--|
| H'FFFF FF80 | TSTRJ    |        | Reserved |          |  |
| H'FFFF FF84 |          | Rese   | rved     |          |  |
| :           |          |        |          |          |  |
| H'FFFF FF8C |          |        |          |          |  |
| H'FFFF FF90 | TCRJ0    | FCRJ0  | TSRJ0    | Reserved |  |
| H'FFFF FF94 | TIERJ0   | FDNRJ0 | NCNTJ0   | NCRJ0    |  |
| H'FFFF FF98 | TCN      | ITJ0   | OCRJ0    |          |  |
| H'FFFF FF9C | FIF      | OJ0    | Reserved |          |  |
| H'FFFF FFA0 | TCRJ1    | FCRJ1  | TSRJ1    | Reserved |  |
| H'FFFF FFA4 | TIERJ1   | FDNRJ1 | NCNTJ1   | NCRJ1    |  |
| H'FFFF FFA8 | TCN      | ITJ1   | OC       | RJ1      |  |
| H'FFFF FFAC | FIFOJ1   |        | Res      | erved    |  |
| H'FFFF FFB0 | Reserved |        |          |          |  |
| :           |          |        |          |          |  |
| H'FFFF FFFC |          |        |          |          |  |

# 13.3 Input/Output Pins

**Table 13.12 Pin Configuration** 

| Block             | Symbol  | I/O    | Function   |
|-------------------|---|--------|--|
| Common controller | TCLKA   | Input  | Input pin for the external clock signal to be supplied on signal line 4 of the clock bus   |
|                   | TCLKB   | Input  | Input pin for the external clock signal that can be supplied on signal line 5 of the clock bus   |
| Timer A           | TIA00 to TIA05  | Input  | Input pins for input-capture triggers for timer A channels   |
| Timer C           | TIOC00 to TIOC03,<br>TIOC10 to TIOC13,<br>TIOC20 to TIOC23,<br>TIOC30 to TIOC33,<br>TIOC40 to TIOC43          | I/O    | Input pins for input-capture triggers and output pins for output-compare signals of timer C (one each for channels 0 to 3 in subblocks C0 to C4) |
| Timer D           | TOD00A to TOD03A,<br>TOD10A to TOD13A,<br>TOD20A to TOD23A,<br>TOD30A to TOD33A                               | Output | Output pins for compare-match signals of timer D (one each for channels 0 to 3 in subblocks D0 to D3)  |
|                   | TOD00B to TOD03B,<br>TOD10B to TOD13B,<br>TOD20B to TOD23B,<br>TOD30B to TOD33B                               | Output | Output pins for one-shot pulses of timer D (one each for channels 0 to 3 in subblocks D0 to D3)  |
| Timer E           | TOE00 to TOE03,<br>TOE10 to TOE13,<br>TOE20 to TOE23,<br>TOE30 to TOE33,<br>TOE40 to TOE43,<br>TOE50 to TOE53 | Output | Output pins for PWM signals of timer E (one each for channels 0 to 3 in subblocks E0 to E5)  |
| Timer F           | TIF0A to TIF2A,<br>TIF3 to TIF19  | Input  | Input pins for event signals for subblocks F0 to F19 of timer F  |
|                   |   |        | TIF0A to TIF2A: Input pins for subblocks F0 to F2  |
|                   |   |        | TIF3 to TIF19: Input pins for subblocks F3 to F19  |
|                   | TIF0B to TIF2B  | Input  | Input pins for event signals for subblocks F0 to F2 of timer F   |
| Timer J           | TIJ0 and TIJ1   | Input  | Input pins for input-capture triggers for subblocks of timer J   |

### 13.4 Overview of Common Controller

The common controller controls the ATU-III module as a whole. For example, it enables and disables the prescalers and timer counters for timers A to J and controls the clock bus.

#### **13.4.1** Clock Bus

The clock bus consists of six signal lines used to distribute the source signals for counting (count enabling signals) to the timer channels. The timer counters on each of the channels run in synchronization with the internal peripheral clock ( $P\phi$ ).

Table 13.13 shows the signals which are available for input on the clock bus.

Table 13.13 Signals to be Input on Clock Bus

| Bit Number of Clock Bus | Input Signals  |
|-------------------------|--|
| 5                       | Output signal from timer B (multiplied-and-corrected clock signal) or external input clock B (TCLKB) |
| 4                       | External input clock A (TCLKA)   |
| 3                       | Output signal from prescaler 3   |
| 2                       | Output signal from prescaler 2   |
| 1                       | Output signal from prescaler 1   |
| 0                       | Output signal from prescaler 0   |

## 13.5 Register Description of Common Controller

#### 13.5.1 ATU-III Master Enable Register (ATUENR)

ATUENR is a 16-bit readable/writable register. This register is used to enable and disable the prescalers and the individual timers in ATU-III. Setting an enable bit to 1 enables the corresponding timer. Clearing the bit to 0 disables the corresponding timer. Even when the enable bit is cleared to 0, the registers of the corresponding timer remain accessible.

Timers can be synchronized by simultaneously setting multiple bits to 1. Note that a particular subblock cannot be synchronized with other subblocks while they are counting.

ATUENR can be read from or written to in byte or word units.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0    |
|----------------|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
|                | -  | -  | -  | -  | -  | -  | TJE | THE | TGE | TFE | TEE | TDE | TCE | TBE | TAE | PSCE |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    |
| R/W:           | R  | R  | R  | R  | R  | R  | R/W  |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 15 to 10 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 9        | TJE      | 0                | R/W | Timer J Enable   |
|          |          |                  |     | Enables and disables counter operation of timer J.   |
|          |          |                  |     | When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer J start register must also be set to 1 to enable the operation of either of the subblock counters. |
|          |          |                  |     | 0: Timer J counter operation disabled  |
|          |          |                  |     | 1: Timer J counter operation enabled   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 8   | THE      | 0                | R/W | Timer H Enable   |
|     |          |                  |     | Enables and disables counter operation of timer H.   |
|     |          |                  |     | When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value.  |
|     |          |                  |     | 0: Timer H counter operation disabled  |
|     |          |                  |     | 1: Timer H counter operation enabled   |
| 7   | TGE      | 0                | R/W | Timer G Enable   |
|     |          |                  |     | Enables and disables counter operation of timer G.   |
|     |          |                  |     | When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer G start register must also be set to 1 to enable the operation of either of the subblock counters. |
|     |          |                  |     | 0: Timer G counter operation disabled  |
|     |          |                  |     | 1: Timer G counter operation enabled   |
| 6   | TFE      | 0                | R/W | Timer F Enable   |
|     |          |                  |     | Enables and disables counter operation of timer F.   |
|     |          |                  |     | When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer F start register must also be set to 1 to enable the operation of either of the subblock counters. |
|     |          |                  |     | 0: Timer F counter operation disabled  |
|     |          |                  |     | 1: Timer F counter operation enabled   |
| 5   | TEE      | 0                | R/W | Timer E Enable   |
|     |          |                  |     | Enables and disables counter operation of timer E.   |
|     |          |                  |     | When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer E start register must also be set to 1 to enable the operation of either of the subblock counters. |
|     |          |                  |     | 0: Timer E counter operation disabled  |
|     |          |                  |     | 1: Timer E counter operation enabled   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 4   | TDE      | 0                | R/W | Timer D Enable   |
|     |          |                  |     | Enables and disables counter operation of timer D.   |
|     |          |                  |     | When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer D start register must also be set to 1 to enable the operation of either of the subblock counters. |
|     |          |                  |     | 0: Timer D counter operation disabled  |
|     |          |                  |     | 1: Timer D counter operation enabled   |
| 3   | TCE      | 0                | R/W | Timer C Enable   |
|     |          |                  |     | Enables and disables counter operation of timer C.   |
|     |          |                  |     | When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer C start register must also be set to 1 to enable the operation of either of the subblock counters. |
|     |          |                  |     | 0: Timer C counter operation disabled  |
|     |          |                  |     | 1: Timer C counter operation enabled   |
| 2   | TBE      | 0                | R/W | Timer B Enable   |
|     |          |                  |     | Enables and disables counter operation of timer B.   |
|     |          |                  |     | When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value.  |
|     |          |                  |     | 0: Timer B counter operation disabled  |
|     |          |                  |     | 1: Timer B counter operation enabled   |
| 1   | TAE      | 0                | R/W | Timer A Enable   |
|     |          |                  |     | Enables and disables counter operation of timer A.   |
|     |          |                  |     | When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value.  |
|     |          |                  |     | 0: Timer A counter operation disabled  |
|     |          |                  |     | 1: Timer A counter operation enabled   |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 0   | PSCE     | 0                | R/W | Prescaler Enable  |
|     |          |                  |     | Enables and disables the prescaler counters. When the prescaler counters are disabled, the counter values are retained. Once the bit is set to 1 again, the counter resumes counting from the retained value. |
|     |          |                  |     | 0: Prescaler counter operation disabled   |
|     |          |                  |     | 1: Prescaler counter operation enabled  |

### 13.5.2 Clock Bus Control Register (CBCNT)

CBCNT is an 8-bit readable/writable register that selects the source of the clock signal to be supplied on signal line 5 of the clock bus and the valid edge of external clock signals (only applies to line 5 when the external input clock is selected).

CBCNT can be read from or written to in bytes.

| Bit:           | 7 | 6 | 5    | 4      | 3 | 2          | 1    | 0      |
|----------------|---|---|------|--------|---|------------|------|--------|
|                | 1 | - | CB4E | G[1:0] | - | CB5<br>SEL | CB5E | G[1:0] |
| Initial value: | 0 | 0 | 0    | 0      | 0 | 0          | 0    | 0      |
| R/W:           | R | R | R/W  | R/W    | R | R/W        | R/W  | R/W    |

| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 7, 6 | _        | All 0            | R   | Reserved  |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 5, 4 | CB4EG    |                  | R/W | Clock Bus 4 Edge Select   |
|      | [1:0]    |                  |     | These bits select the edge sense for external input clock A (TCLKA). The clock signal is output on signal line 4 of the clock bus. Counters for which signal line 4 of the clock bus has been selected as the source for counting count on the edge selected by these bits. |
|      |          |                  |     | 00: Neither edge of the external clock is sensed  |
|      |          |                  |     | 01: Rising edges of the external clock are sensed   |
|      |          |                  |     | 10: Falling edges of the external clock are sensed  |
|      |          |                  |     | <ol> <li>Both rising and falling edges of the external clock are<br/>sensed</li> </ol>  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 3    | _        | 0                | R   | Reserved  |
|      |          |                  |     | This bit is always read as 0. The write value should always be 0.   |
| 2    | CB5SEL   | 0                | R/W | Clock Bus 5 Source Select   |
|      |          |                  |     | Selects the source of the clock to be output on signal line 5 of the clock bus.   |
|      |          |                  |     | 0: External input clock B (TCLKB)   |
|      |          |                  |     | 1: Multiplied-and-corrected clock output by Timer B   |
| 1, 0 | •        | 00               | R/W | Clock Bus 5 Edge Select   |
|      | [1:0]    |                  |     | These bits select the edge sense for external input clock B (TCLKB). The clock signal is output on signal line 5 of the clock bus. Counters for which signal line 5 of the clock bus has been selected as the source for counting count on the edge selected by these bits. The setting of these bits is only valid when the TCLKB signal is selected as the source for line 5 of the clock bus. When the multiplied-and-corrected clock is selected as the source for line 5 of the clock bus, the setting of these bits is invalid. |
|      |          |                  |     | 00: Neither edge of the external clock is sensed  |
|      |          |                  |     | 01: Rising edges of the external clock are sensed   |
|      |          |                  |     | 10: Falling edges of the external clock are sensed  |
|      |          |                  |     | 11: Both rising and falling edges of the external clock are sensed  |

### 13.5.3 Noise Cancellation Mode Register (NCMR)

NCMR is an 8-bit readable/writable register that selects the mode and clock to drive the counter for of the noise canceler in each of timers A, C, F, and J.

In premature-transition cancellation mode, subsequent changes to the input signal level are ignored if they come within a given period of a detected change. That is, level changes within a certain period of an initial one are treated as noise.

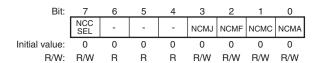
In minimum time-at-level cancellation mode, the first and subsequent level changes are ignored unless the input signal level remains the same over a given period. Level changes occurring within a shorter period are considered to indicate an unstable signal, and such signals are treated as noise.

The period is set by noise canceler registers in each of the applicable blocks (i.e. in timers A, C, F, and J) and is counted by a noise canceler counter.

Figures 13.1 and 13.2 show the operation of the premature-transition cancellation and minimum time-at-level cancellation, using the TIA00 input signal of timer A as an example.

The edge for counting is detected from signals after noise removal in timers A, C, F, and J. Rising edges are being detected in figures 13.1 and 13.2.

NCMR can be read from and written to in byte units.



Page 394 of 1824

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7      | NCCSEL   | 0                | R/W | Noise Canceler Counter Clock Select  |
|        |          |                  |     | Selects the clock for counting by the noise cancelers. The peripheral clock ( $P\phi$ ) or $P\phi$ divided by 128 can be selected. The default setting is the clock divided by 128. The same counter clock must be used for all timers other than timer A. In the case of timer A, the clock signal on clock-bus line 5 is also available. For details, see section 13.10.3, Timer I/O Control Register 2A (TIOR2A). |
|        |          |                  |     | 0: P $\phi$ divided by 128 is used as the counter clock  |
|        |          |                  |     | 1: Pφ is used as the counter clock   |
| 6 to 4 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 3      | NCMJ     | 0                | R/W | Timer J Noise Cancellation Mode  |
|        |          |                  |     | Selects the noise cancellation mode for timer J. The same mode is used on both channels of timer J.  |
|        |          |                  |     | 0: Premature-transition cancellation mode  |
|        |          |                  |     | 1: Minimum time-at-level cancellation mode   |
| 2      | NCMF     | 0                | R/W | Timer F Noise Cancellation Mode  |
|        |          |                  |     | Selects the noise cancellation mode for timer F. The same mode is used on both channels of timer F.  |
|        |          |                  |     | 0: Premature-transition cancellation mode  |
|        |          |                  |     | 1: Minimum time-at-level cancellation mode   |
| 1      | NCMC     | 0                | R/W | Timer C Noise Cancellation Mode  |
|        |          |                  |     | Selects the noise cancellation mode for timer C. The same mode is used on both channels of timer C.  |
|        |          |                  |     | 0: Premature-transition cancellation mode  |
|        |          |                  |     | 1: Minimum time-at-level cancellation mode   |
| 0      | NCMA     | 0                | R/W | Timer A Noise Cancellation Mode  |
|        |          |                  |     | Selects the noise cancellation mode for timer A. The same mode is used on both channels of timer A.  |
|        |          |                  |     | 0: Premature-transition cancellation mode  |
|        |          |                  |     | 1: Minimum time-at-level cancellation mode   |

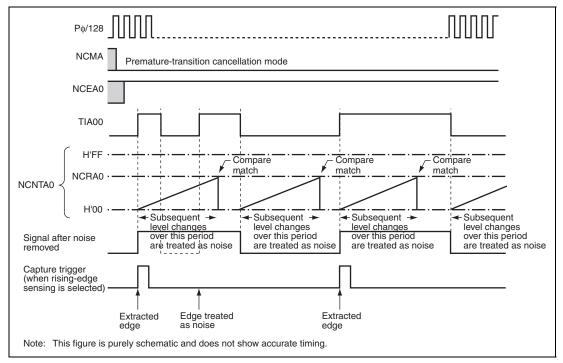


Figure 13.1 Operation of Noise Canceler in Premature-Transition Cancellation Mode

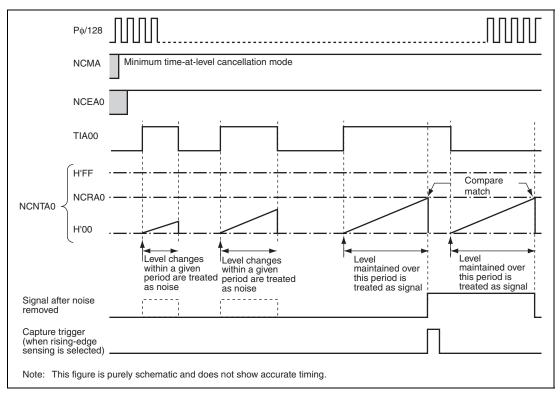


Figure 13.2 Operation of Noise Canceler in Minimum-Time-at Level Cancellation Mode

#### 13.6 Overview of Prescalers

ATU-III includes four general prescalers and a prescaler for the noise-canceler clock.

The general prescalers are implemented as 10-bit down-counters, in which the prescaled clock signals are derived by frequency-dividing the peripheral clock (P $\phi$ ) by N (1  $\leq$  N  $\leq$  1024).

The division ratio is obtained from the following expression.

Division ratio of prescaler = 
$$\frac{1}{PSCn[9:0] + 1}$$
 (Settable value: 1/1 to 1/1024)

A duty cycle of 50% is not guaranteed for the clock-signal outputs of the prescalers. Instead, the high level is output in one cycle of the  $P\phi$  clock and the low level is output in the remaining cycles within the prescaled period. When 1/1 is selected as the division ratio, the high level is always output on the clock bus.

The generated clock signals are supplied to the individual timers via the clock bus. The prescalers for each of the channels operate independently. The prescalers can, however, be started in synchronization with each other after a reset by setting the PSCE bit in ATUENR to 1 after the appropriate values have been set. Synchronization of a prescaler is not possible after it has started or its division ratio has been changed.

The prescaler for the noise-canceler clock is implemented as a 7-bit down-counter. This generates a clock signal by frequency-dividing the peripheral clock ( $P\phi$ ) by 128. The clock signal thus generated is supplied to the noise cancelers of timers A, C, F, and J.

The peripheral clock of the peripheral clock frequency divided by 128 can be selected for the noise-canceler clock by the NCCSEL bit in NCMR of the common controller. Further division ratios are not available.

The down-counters of the prescalers are initialized to H'000 by a power-on reset or a transition to the hardware standby mode.

## 13.7 Register Description of Prescalers

#### 13.7.1 Prescaler Registers 0 to 3 (PSCR0 to PSCR3)

PSCR0 to PSCR3 are 16-bit readable/writable registers, each of which holds a division ratio for one of the four prescalers.

After a prescaler counter underflows, counting restarts from the setting in this register. Settable values range from H'000 to H'3FF.

PSCR0 to PSCR3 can be read from or written to in word units.

| Bit:           | 15       | 14  | 13 | 12 | 11 | 10 | 9   | 8   | 7   | 6   | 5   | 4      | 3   | 2   | 1   | 0   |
|----------------|----------|-----|----|----|----|----|-----|-----|-----|-----|-----|--------|-----|-----|-----|-----|
|                | 1        | -   | ı  | ı  | -  | -  |     |     |     |     | PSC | ո[9:0] |     |     |     |     |
| Initial value: | 0        | 0   | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0      | 0   | 0   | 0   | 0   |
| R/W:           | R        | R   | R  | R  | R  | R  | R/W | R/W | R/W | R/W | R/W | R/W    | R/W | R/W | R/W | R/W |
| Note:          | n = 0 to | 0 3 |    |    |    |    |     |     |     |     |     |        |     |     |     |     |

|          |           | Initial |     |  |
|----------|-----------|---------|-----|--|
| Bit      | Bit Name  | Value   | R/W | Description  |
| 15 to 10 | _         | All 0   | R   | Reserved   |
|          |           |         |     | These bits are always read as 0. The write value should always be 0.   |
| 9 to 0   | PSCn[9:0] | All 0   | R/W | Division Ratio   |
|          |           |         |     | These bits specify the division ratio for the corresponding prescaler. |

## 13.8 Operation of Prescalers

## 13.8.1 Starting Prescalers

The prescalers start operating when the PSCE bit in the ATU-III master enable register (ATUENR) is set to 1 and generates a clock with a frequency given by the division ratio in the PSCn bits. While a prescaler is operating, the high level is output for one cycle of the  $P\phi$  clock each time the corresponding prescaler counter underflows.

When the setting in the PSCn bits is changed during operation, the division ratio of the output clock is updated on the first subsequent counter underflow.

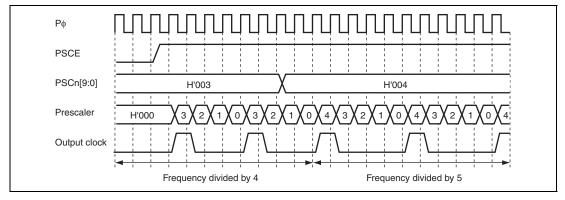


Figure 13.3 Starting Prescaler

#### 13.8.2 Stopping and Restarting Operation

The prescaler stops operating when the PSCE bit in the ATU-III master enable register (ATUENR) is cleared to 0. The clock signal stays at the low level and the value in the prescaler counter is retained while the prescaler is stopped. Setting the PSCE bit to 1 makes counting restart from the retained value.

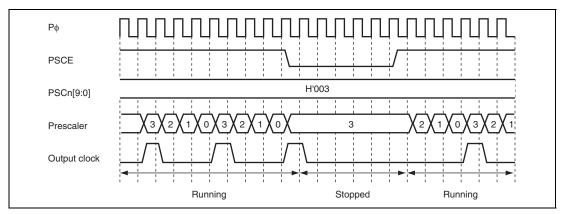


Figure 13.4 Stopping Prescaler

#### 13.9 Overview of Timer A

Timer A includes a free-running counter A (TCNTA) and input capture registers A0 to A5 (ICRA0 to ICRA5). TCNTA is a free-running up counter. An interrupt request can be output when the counter overflows.

Input capture registers A0 to A5 (ICRA0 to ICRA5) capture the value of free-running counter A (TCNTA) on an assertion of the corresponding external input signal (TIA00 to TIA05). The rising or falling edge, or both edges, can be selected as the trigger for capture. The edge selection is made timer I/O control register 1A. The interrupt is requested or the A-DMAC is activated at the same timing as capturing.

Noise on the external input signals can be removed by the noise canceler. Signals on pins TIA00 to TIA02 can be output to timer B or D as event signals after noise has been removed and their edges have been extracted. One of TIA00 to TIA02 are selectable for output to timer B as an event signal (event output 1). TIA01 and TIA02 are selectable for output to timers D0 to D3 (for timers D0 to D2 as event output 2A and for timer D3 as event output 2B) and these event outputs can be used as capture triggers for timer offset base registers for D0 to D3 (OSBRD0 to OSBRD3).

When the A-DMAC is activated by a capture operation, it reads the capture register associated with the interrupt source and transfers the value read from the register and any previously captured value stored in the A-DMAC buffer register to the on-chip RAM.

# 13.9.1 Block Diagram of Timer A

Figure 13.5 is a block diagram of timer A.

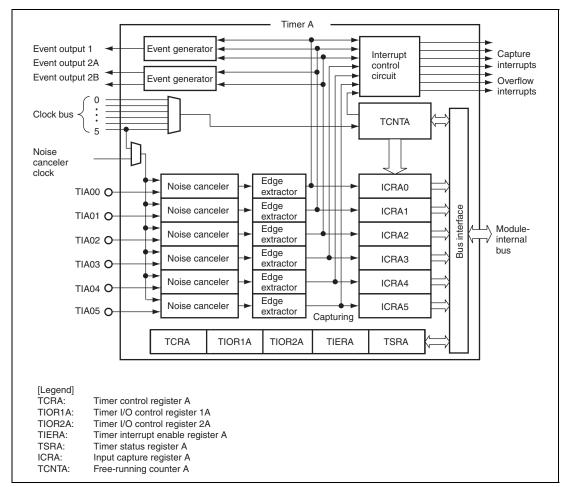


Figure 13.5 Block Diagram of Timer A

# 13.10 Description of Timer A Registers

# 13.10.1 Timer Control Register A (TCRA)

TCRA is an 8-bit readable/writable register that sets the event output generated from external input signals (TIA00 to TIA02) and the counter clock.

TCRA can be read from and written to in byte or word units.

TCRA is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7            | 6            | 5   | 4     | 3   | 2      | 1   | 0   |  |
|----------------|--------------|--------------|-----|-------|-----|--------|-----|-----|--|
|                | EVO<br>SEL2A | EVO<br>SEL2B | Е   | VOSEL | .1  | CKSELA |     |     |  |
| 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<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 7   | EVOSEL2A | 0                | R/W | Event Output 2A Select   |
|     |          |                  |     | Selects the signal for output on event output 2A as the externally input signal TIA01 or TIA02 (after the removal of noise and edge extraction). In timers D0 to D2, the event output 2A signal can be used as a capture trigger for the corresponding timer-offset base registers (OSBRD0 to OSBRD2). |
|     |          |                  |     | 0: TIA01 is selected as event output 2A  |
|     |          |                  |     | 1: TIA02 is selected as event output 2A  |
| 6   | EVOSEL2B | 0                | R/W | Event Output 2B Select   |
|     |          |                  |     | Selects the signal for output on event output 2B as the externally input signal TIA01 or TIA02 (after the removal of noise and edge extraction). In timer D3, the event output 2B signal can be used as a capture trigger for the corresponding timer-offset base register (OSBRD3).                   |
|     |          |                  |     | 0: TIA01 is selected as event output 2B  |
| 1   |          |                  |     | 1: TIA02 is selected as event output 2B  |

|          | Initial |                             |  |
|----------|---------|-----------------------------|--|
| Bit Name | Value   | R/W                         | Description  |
| EVOSEL1  | 000     | R/W                         | Event Output 1 Select  |
|          |         |                             | These bits select the signal for output on event output 1 as one of the externally input signal TIA00 to TIA02 (after the removal of noise and edge extraction). In timer B, the event output 1 signal can be used as an event output 1 to timer B.  |
|          |         |                             | 000: No signal is selected as event output 1   |
|          |         |                             | 001: TIA00 is selected as event output 1   |
|          |         |                             | 010: TIA01 is selected as event output 1   |
|          |         |                             | 011: Setting prohibited  |
|          |         |                             | 100: TIA02 is selected as event output 1   |
|          |         |                             | 101: Setting prohibited  |
|          |         |                             | 110: Setting prohibited  |
|          |         |                             | 111: Setting prohibited  |
| CKSELA   | 000     | R/W                         | Clock Select A   |
|          |         |                             | These bits select the signal on one of clock-bus lines 0 to 5 as the clock signal for counting. The signal on lines 0 to 3 have been frequency-divided by prescalers 0 to 3, respectively. Clock-bus line 4 supplies externally input clock A (TCLKA). Clock-bus line 5 supplies externally input clock B (TCLKB) or the multiplied-and-corrected clock output by timer B. |
|          |         |                             | Stop timer A before making or changing the counter-clock selection.  |
|          |         |                             | 000: Clock-bus line 0 (prescaler 0)  |
|          |         |                             | 001: Clock-bus line 1 (prescaler 1)  |
|          |         |                             | 010: Clock-bus line 2 (prescaler 2)  |
|          |         |                             | 011: Clock-bus line 3 (prescaler 3)  |
|          |         |                             | 100: Clock-bus line 4 (TCLKA)  |
|          |         |                             | <ol> <li>Clock-bus line 5 (TCLKB or multiplied-and-corrected clock)</li> </ol>   |
|          |         |                             | 110: Setting prohibited  |
|          |         |                             | 111: Setting prohibited  |
|          |         | Bit Name Value  EVOSEL1 000 | Bit Name Value R/W EVOSEL1 000 R/W   |

Note: The edge of an external input clock is extracted before it is output on a clock bus. When using external input clock A or B, select the edge to be extracted by setting the CB4EG and CB5EG bits in the clock bus control register (CBCNT).

#### 13.10.2 Timer I/O Control Register 1A (TIOR1A)

TIOR1A is a 16-bit readable/writable register that sets the edge of external inputs (TIA00 to TIA05) to be extracted.

TIOR1A can be read from and written to in byte or word unit.

TIOR1A is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 15 | 14 | 13 | 12 | 11  | 10         | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|----------------|----|----|----|----|-----|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                | -  | -  | -  | -  | 10  | <b>A</b> 5 | Ю   | A4  | Ю   | A3  | Ю   | A2  | Ю   | A1  | Ю   | A0  |
| Initial value: | 0  | 0  | 0  | 0  | 0   | 0          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R  | R  | R  | R  | R/W | R/W        | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 15 to 12 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 11, 10   | IOA5     | 00               | R/W | I/O Control An   |
| 9, 8     | IOA4     | 00               | R/W | These bits select the edge of external inputs (TIA00 to  |
| 7, 6     | IOA3     | 00               | R/W | TIA05) that is to be extracted for use in input-capture  |
| 5, 4     | IOA2     | 00               | R/W | <ul> <li>triggering. When these bits are set to B'00, input capturing</li> <li>is not performed. When a value other than B'00 is set, the</li> </ul> |
| 3, 2     | IOA1     | 00               | R/W | contents of free-running counter A (TCNTA) are   |
| 1, 0     | IOA0     | 00               | R/W | transferred to input capture register A (ICRA) on extraction of the selected edge from one of the external inputs.                                   |

sure that the frequency of the Po clock is at least twice the frequency of the external input signal. Otherwise, edge extraction will not performed correctly.

When the noise canceler is disabled, the selected edge is simply extracted from the external inputs (TIA00 to TIA05). When the noise canceler is enabled, the selected edge is extracted from the signals after noise removal.

Usage of the extracted edges in other timers is selected by the settings of bits EVOSEL1, EVOSEL2A, and EVOSEL2B in TCRA (the signal outputs for other timers as a result of edge extraction are active high).

00: Input capturing is not performed

01: TCNTA is captured in ICRA on the rising edge of TIA

TCNTA is captured in ICRA on the falling edge of TIA

11: TCNTA is captured in ICRA on both edges of TIA

Note: n = 0 to 5

# 13.10.3 Timer I/O Control Register 2A (TIOR2A)

TIOR2A is a 16-bit readable/writable register that selects the noise canceler clock and enables and disables the noise cancelers for externally input signals (TIA00 to TIA05).

TIOR2A can be read from and written to in byte or word unit.

TIOR2A is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 15 | 14 | 13    | 12    | 11    | 10    | 9     | 8     | 7 | 6 | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|-------|-------|-------|-------|-------|-------|---|---|-------|-------|-------|-------|-------|-------|
|                | -  | -  | NCKA5 | NCKA4 | NCKA3 | NCKA2 | NCKA1 | NCKA0 | - | - | NCEA5 | NCEA4 | NCEA3 | NCEA2 | NCEA1 | NCEA0 |
| Initial value: | 0  | 0  | 0     | 0     | 0     | 0     | 0     | 0     | 0 | 0 | 0     | 0     | 0     | 0     | 0     | 0     |
| B/W·           | R  | R  | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R | R | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 15, 14 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 13     | NCKA5    | 0                | R/W | Noise Canceler Clock Select An   |
| 12     | NCKA4    | 0                | R/W | These bits select the signal that drives counting. The noise   |
| 11     | NCKA3    | 0                | R/W | canceler count clock or the signal on clock-bus line 5 can be selected as the signal for counting. Either the Pφ clock |
| 10     | NCKA2    | 0                | R/W | frequency divided by 128 or the P $\phi$ clock can be selected as  |
| 9      | NCKA1    | 0                | R/W | the noise canceler count clock by setting the NCCSEL bit of the common controller.                                     |
| 8      | NCKA0    | 0                | R/W | O: Noise canceler count clock is selected as the signal for counting by NCNTAn   |
|        |          |                  |     | 1: Clock-bus line 5 is selected as the signal for counting by NCNTAn   |
| 7, 6   | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |

|     |          | Initial |     |            |
|-----|----------|---------|-----|------------|
| Bit | Bit Name | Value   | R/W | Descr      |
| 5   | NCEA5    | 0       | R/W | Noise      |
| 4   | NCEA4    | 0       | R/W | These      |
| 3   | NCEA3    | 0       | R/W | extern     |
| 2   | NCEA2    | 0       | R/W | When TIA05 |
| 1   | NCEA1    | 0       | R/W | prema      |
| 0   | NCEA0    | 0       | R/W | cancel     |

Description

Noise Canceler Enable An

These bits enable and disable the noise cancelers for externally input signals (TIA00 to TIA05)

When a level change on externally input signals TIA00 to TIA05 is detected while this bit is set to 1, it is processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the noise cancellation mode register (NCMR) of the common controller.

0: Noise cancelers for TIA are disabled

1: Noise cancelers for TIA are enabled

• In premature-transition cancellation mode

When a level change of the externally input signal is detected, the change is output as the signal whose noise is removed and the corresponding noise canceler counter (NCNTA0 to NCNTA5) is started for counting up. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRA0 to NCRA5). The level of the externally input signal is output on this compare match.

When these bits are cleared to 0 while the counter (NCNTA0 to NCNTA5) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

In minimum time-at-level cancellation mode

When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTA0 to NCNTA5) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRA0 to NCRA5), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed.

When these bits are cleared to 0 while the counter (NCNTA0 to NCNTA5) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected.

For details on operations in each mode, see figures 13.1 and 13.2.

### 13.10.4 Timer Status Register A (TSRA)

TSRA is an 8-bit readable/writable register that indicates overflow on free-running counter A (TCNTA) and input capture on input capture registers A0 to A5 (ICRA0 to ICRA5).

The flags in this register are used as interrupts. When the corresponding bit in timer interrupt enable register A (TIERA) is enabled, setting of a flag can lead to a request DMA transfer by A-DMAC or sending of an interrupt request to the CPU via the A-DMAC.

TSRA can be read from and written to in byte or word units.

TSRA is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Note: \* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |
|-----|----------|------------------|--------|---|
| 7   | OVFA     | 0                | R/(W)* | Overflow Flag A   |
|     |          |                  |        | Indicates that free-running counter A (TCNTA) has overflowed.   |
|     |          |                  |        | This flag cannot be set to 1 by software. To clear this bit, write 0 to it after reading it as 1. Writing 0 before reading it as 1 has no effect. |
|     |          |                  |        | 0: Indicates that TCNTA has not overflowed  |
|     |          |                  |        | [Clearing condition]  |
|     |          |                  |        | When writing 0 to this bit after reading it as 1  |
|     |          |                  |        | 1: Indicates that TCNTA has overflowed  |
|     |          |                  |        | [Setting condition]   |
|     |          |                  |        | <ul> <li>When TCNTA overflows (transition from<br/>H'FFFFFFF to H'00000000)</li> </ul>  |
| 6   | _        | 0                | R      | Reserved  |
|     |          |                  |        | This bit is always read as 0. The write value should always be 0.   |

|     |          | Initial |        |  |
|-----|----------|---------|--------|--|
| Bit | Bit Name | Value   | R/W    | Description  |
| 5   | ICFA5    | 0       | R/(W)* | Input Capture Flag An  |
| 4   | ICFA4    | 0       | R/(W)* | These bits indicate that the value in free-running   |
| 3   | ICFA3    | 0       | R/(W)* | counter A (TCNTA) has been captured by input capture register An (ICRAn). When one of these bits                               |
| 2   | ICFA2    | 0       | R/(W)* | is read as 1, the value in TCNTA has been stored in  |
| 1   | ICFA1    | 0       | R/(W)* | the corresponding ICRAn.   |
| 0   | ICFA0    | 0       | R/(W)* | These bits are not set to 1 by software.   |
|     |          |         |        | These bits are automatically cleared to 0 when ICRAn is read by the A-DMAC. Accesses by the CPU, AUD, and DMAC have no effect. |
|     |          |         |        | Each bit is cleared by writing 0 to it after reading as 1. Writing 0 before reading it as 1 has no effect.                     |
|     |          |         |        | 0: No input capture has occurred   |
|     |          |         |        | [Clearing conditions]  |
|     |          |         |        | When writing 0 to this bit after reading it as 1   |
|     |          |         |        | When ICRAn is read by A-DMAC   |
|     |          |         |        | 1: Input capture has occurred  |
|     |          |         |        | [Setting condition]  |
|     |          |         |        | When TCNTA is transferred to ICRAn on assertion<br>of the input capture signal (TIA0n)   |

<sup>\*</sup> Only writing 0 to this bit after reading it as 1 to clear the flag. Writing 1 to this bit has no effect.

# 13.10.5 Timer Interrupt Enable Register A (TIERA)

TIERA is an 8-bit readable/writable register that enables and disables an interrupt request of overflow on free-running counter A (TCNTA) or input capture on input capture registers A0 to A5 (ICRA0 to ICRA5). The input capture interrupt request can be used to request DMA transfer between input capture registers and the on-chip RAM according to the settings in the A-DMAC.

TIERA can be read from and written to in byte or word units.

TIERA is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7    | 6 | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|------|---|-------|-------|-------|-------|-------|-------|
|                | OVEA | - | ICEA5 | ICEA4 | ICEA3 | ICEA2 | ICEA1 | ICEA0 |
| Initial value: | 0    | 0 | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R/W  | R | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |

| Bit | Bit Name | Initial<br>Value | D/M | Description   |
|-----|----------|------------------|-----|---|
| DIL | bit Name | value            | R/W | Description   |
| 7   | OVEA     | 0                | R/W | Overflow Interrupt A Enable   |
|     |          |                  |     | Enables and disables an OVFA interrupt request when overflow flag A (OVFA) in timer status register A (TSRA) is set to 1. |
|     |          |                  |     | 0: Disables an OVFA interrupt request   |
|     |          |                  |     | 1: Enables an OVFA interrupt request  |
| 6   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.   |
| 5   | ICEA5    | 0                | R/W | Input Capture Interrupt An Enable   |
| 4   | ICEA4    | 0                | R/W | Enables and disables an ICFAn interrupt request when bit  |
| 3   | ICEA3    | 0                | R/W | ☐ ICFAn in TSRA is set to 1. When the interrupt is enabled,<br>_ it can be used to request DMAC transfer by the A-DMAC.   |
| 2   | ICEA2    | 0                | R/W | _0: Disables a request of input capture interrupt An  |
| 1   | ICEA1    | 0                | R/W | _ 1: Enables a request of input capture interrupt An  |
| 0   | ICEA0    | 0                | R/W |   |

### 13.10.6 Input Capture Registers A0 to A5 (ICRA0 to ICRA5)

ICRA0 to ICRA5 are 32-bit read-only registers used for input capturing. Writing to these registers is prohibited.

These registers hold the contents of free-running counter A (TCNTA) when an assertion of input capture signals (TIA00 to TIA05) is detected. At this time, the ICFA bit in timer status register A (TSRA) is set to 1. The bit is cleared to 0 when ICRA0 to ICRA5 are read by the A-DMAC.

The edge to be detected is selected by bit I/O control An in timer I/O control register1A (TIOR1A).

ICRA0 to ICRA5 can be read from in longword units.

ICRA0 to ICRA5 are initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24    | 23     | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|----|----|----|----|----|----|----|-------|--------|----|----|----|----|----|----|----|
|                |    |    |    |    |    |    |    | ICAn[ | 31:16] |    |    |    |    |    |    |    |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R  | R  | R  | R  | R  | R  | R     | R      | R  | R  | R  | R  | R  | R  | R  |
| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8     | 7      | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|                |    |    |    |    |    |    |    | ICAn  | [15:0] |    |    |    |    |    |    |    |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R  | R  | R  | R  | R  | R  | R     | R      | R  | R  | R  | R  | R  | R  | R  |

| Bit     | Initial<br>Bit Name Value | R/W | Description                                 |
|---------|---------------------------|-----|---|
| 31 to 0 | ICAn[31:0] All 0          | R   | Input Capture An                            |
|         |                           |     | These bits hold 32-bit input capture value. |

## 13.10.7 Free-Running Counter A (TCNTA)

TCNTA is a 32-bit readable/writable register that counts on the signal output by the prescaler via the clock bus, externally input clock signal, or multiplied-and-corrected signal output by timer B.

Timer A is started for counting up by setting the TAE bit in the ATU-III master enable register (ATUENR) to 1. The clock input to the counter is selected by setting the clock select bit (CKSELA) in timer control register A (TCRA).

When TCNTA overflows (from H'FFFF FFFF to H'0000 0000), the overflow flag A (OVFA) in timer status register A (TSRA) is set to 1.

TCNTA can be read from and written to in longword units.

TCNTA is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24   | 23     | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|------|--------|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     | CNTA | 31:16] |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0      | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W  | R/W    | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8    | 7      | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     | CNTA | [15:0] |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0      | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W  | R/W    | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 31 to 0 | CNTA     | All 0            | R/W | Timer Count A                                      |
|         | [31:0]   |                  |     | These bits hold the counter value in 32-bit units. |

### 13.10.8 Noise Canceler Counters A0 to A5 (NCNTA0 to NCNTA5)

NCNTA0 to NCNTA5 are an 8-bit readable/writable registers that are started for counting up by an assertion of externally input signals (TIA00 to TIA05) as a trigger when the noise cancelers are enabled by setting the noise canceler enable bit (NCEA5 to NCEA0) in timer I/O control register 2A (TIOR2A) to 1. These counters are driven by the clock selected in the noise canceler clock select bits (NCKA5 to NCKA0). The counter clock for noise cancelers or clock-bus line 5 are available.

Input edges are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the NCMA bit in the noise cancellation mode register (NCMR) of the common controller.

## • Premature-Transition Cancellation Mode

When a level change of the externally input signal (TIA00 to TIA05) is detected while bits NCEA5 to NCEA0 are set to 1 and NCNTA0 to NCNTA5 are stopped, NCNTA0 to NCNTA5 are started for counting up. These counters are cleared to H'00 and stopped on the first edge of the P $\varphi$  clock after the value in NCNTA0 to NCNTA5 matches the value in noise canceler registers A0 to A5 (NCRA0 to NCRA5).

NCNTA0 to NCNTA5 are incremented regardless of the TAE bits in ATUENR.

The first change is output as the signal whose noise is removed and the edge is to be extracted. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRA0 to NCRA5). The level of the externally input signal is output on this compare match.

When the NCEA bits are cleared to 0 while the counter (NCNTA0 to NCNTA5) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

### • Minimum Time-at-Level Cancellation Mode

When a level change of the externally input signal (TIA00 to TIA05) is detected while bits NCEA5 to NCEA0 are set to 1 and NCNTA0 to NCNTA5 are stopped, NCNTA0 to NCNTA5 are started for counting up. These counters are cleared to H'00 and stopped on the first edge of the  $P\phi$  clock after the value in NCNTA0 to NCNTA5 matches the value in noise canceler registers A0 to A5 (NCRA0 to NCRA5) or after the level of the externally input signal (TIA00 to TIA05) is changed.

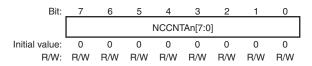
NCNTA0 to NCNTA5 are incremented regardless of the TAE bits in ATUENR.

When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTA0 to NCNTA5) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRA0 to NCRA5), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed.

When the NCEA bits are cleared to 0 while the counter (NCNTA0 to NCNTA5) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected.

NCNTA5 to NCNTA0 can be read from and written to in byte or word units.

NCNTA5 to NCNTA0 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



| Bit    | Bit Name                       | Initial<br>Value | R/W | Description                                      |  |
|--------|--------------------------------|------------------|-----|--|--|
| 7 to 0 | 0 0 NCCNTAn All 0 R/W<br>[7:0] |                  | R/W | Noise Canceler Count An                          |  |
|        |                                |                  |     | These bits hold the counter value in byte units. |  |

## 13.10.9 Noise Canceler Registers A0 to A5 (NCRA0 to NCRA5)

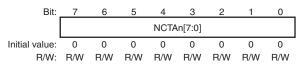
NCRA0 to NCRA5 are 8-bit readable/writable registers that set the upper limitations of noise canceler counters (NCNTA0 to NCNTA5). For example, when the noise canceler is driven by the  $P\phi$  clock divided by 128 and these registers are set to H'FF, a pulse whose width is 1.64 ms (P( is 20 MHz) can be treated as noise.

Input edges are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the NCMA bit in the noise cancellation mode register (NCMR) of the common controller.

For details on these modes, see section 13.10.8, Noise Canceler Counters A0 to A5 (NCNTA0 to NCNTA5).

NCRA0 to NCRA5 can be read from and written to in byte or word units.

NCRA0 to NCRA5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7 to 0 | NCTAn    | All 0            | R/W | Noise Cancellation Time An                                    |
|        | [7:0]    |                  |     | These bits set a period for noise cancellation in byte units. |

# 13.11 Operations of Timer A

## 13.11.1 Operation of Noise Canceler

Input edges are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the NCMA bit in the noise cancellation mode register (NCMR) of the common controller.

Figures 13.6 and 13.8 show examples of noise cancellation in premature-transition cancellation and minimum time-at-level cancellation modes, respectively. In these examples, edges are input on pin TIA00 and the rising edge sensing is selected.

In premature-transition cancellation mode, noise canceler counter A (NCNTA) is started by the level change of the externally input signal as a trigger. At the same time, the level change is output as the signal after noise removal.

Counting continues until the counter value match the value in noise canceler register A (NCRA). Level changes within the period are ignored and are not output. When values in the counter and NCRA match, the level of the externally input signal is output. Note that the levels are changed on the compare match when the input levels at the start of counting (after the first level change) and on the compare match differ.

Figure 13.7 shows an example of noise cancellation for two types of waveforms.

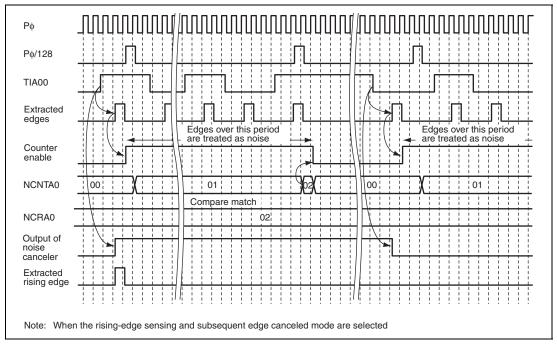


Figure 13.6 Example of Noise Cancellation in Premature-Transition Cancellation

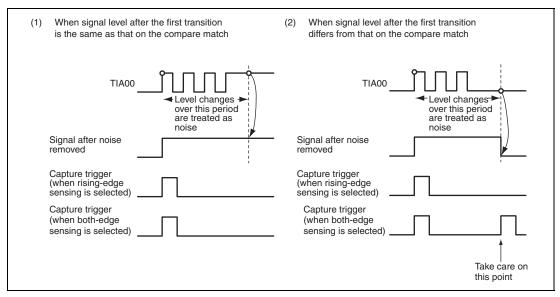


Figure 13.7 Example of Noise Cancellation in Premature-Transition Cancellation for Two
Types of Waveforms

In minimum time-at-level cancellation, noise canceler counter (NCNTA) is started by the level change of the externally input signal as a trigger. Counting continues until the counter value match the value in noise canceler register A (NCRA) or the level change of the input signal is detected.

When the values in the counter and noise canceler register A (NCRA) match, the level change at the start of counting is output as the signal after noise removal. When the level change is detected before the compare match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed.

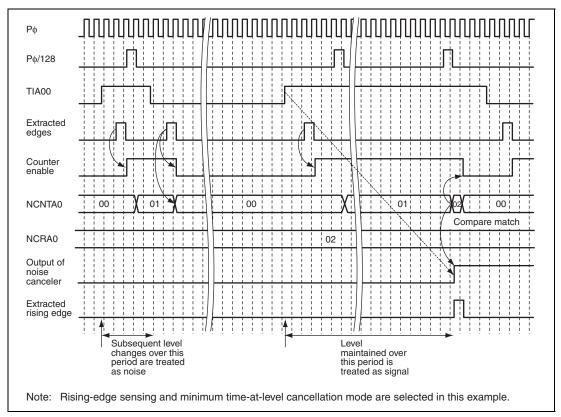


Figure 13.8 Example of Noise Cancellation in Minimum Time-at-Level Cancellation

## 13.11.2 Operation of Free-Running Counter

Free-running counter A (TCNTA) is started for counting up by setting the TAE bit in ATU-III master enable register (ATUENR) to 1. When TCNTA overflows (from H'FFFFFFFF to H'00000000), the OVFA bit in timer status register A (TSRA) is set to 1. An interrupt request is issued for the CPU when the OVEA bit in timer interrupt enable register A (TIERA) is set to 1. After overflow, TCNTA continues counting up from H'00000000.

When the TAE bit in AUTENR is cleared to 0, TCNTA is stopped but is not cleared. By setting the TAE bit to 1 again, TCNTA is resumed from the value when stopped.

TCNTA can be written during operation and writing takes priority over counting. After that, TCNTA is started from the written value. Regardless of the clock driving the counter, the write access is completed in two cycles of the  $P\phi$  clock.

The prescalers run regardless of the TAE bit and are not synchronized with the timing at which the TAE bit is set. Therefore, the time from when the TAE bit is set to when TCNTA is incremented for the first time is less than the cycle of the clock of TCNTA.

Figure 13.9 shows an example of free-running counter A (TCNTA) operation.

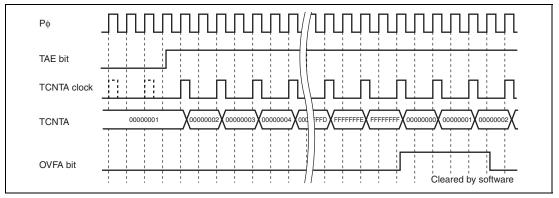


Figure 13.9 Example of Free-Running Counter A (TCNTA) Operation: Overflow Timing

## 13.11.3 Input Capture

Input capture is performed by input capture registers A0 to A5 (ICRA0 to ICRA5) when input capture is enabled in bits IOA5 to IOA0 in timer I/O control register 1A (TIOR1A). ICRA0 to ICRA5 capture the value in free-running counter A (TCNTA) by an edge of externally input signals (TIA00 to TIA05). Noise on the signals can be removed by the noise cancelers.

TCNTA is started for counting up by setting the TAE bit in AUTENR. When an edge is input on the corresponding signal, the bit in timer status register A (TSRA) is set to 1 and the value in TCNTA is transferred to ICRA. The rising or falling edge, or both edges can be selected. Interrupt requests can be issued for the CPU by setting interrupt enable register A (TIERA) and DMA transfer by the A-DMAC can be also activated by the interrupt requests.

When input capture registers A0 to A5 (ICRA0 to ICRA5) and free-running counter A (TCNTA) are written simultaneously, ICRA0 to ICRA5 capture the previous value stored in TCNTA.

Figure 13.10 shows an example of input capture when the edges to be sensed are rising edges for TIA00, falling edges for TIA01, and both edges for TIA02.

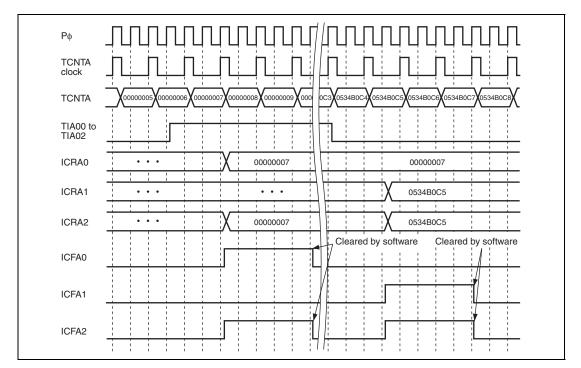


Figure 13.10 Example of Input Capture of Timer A

Signals after noise cancellation and edge extraction can be output to timer B by setting bits EVOSEL1[2:0] in timer control register A (TCRA). By setting bits EVOSEL2A or EVOSEL2B, TIOA01 and TIA02 after noise cancellation and edge extraction can be output to timer D as events.

Figure 13.11 shows an example of event output when TIA00 is selected as event output 1 and TIA01 is selected as event output 1 (EVOSEL1[2:0] = B'001 and B'010, respectively).

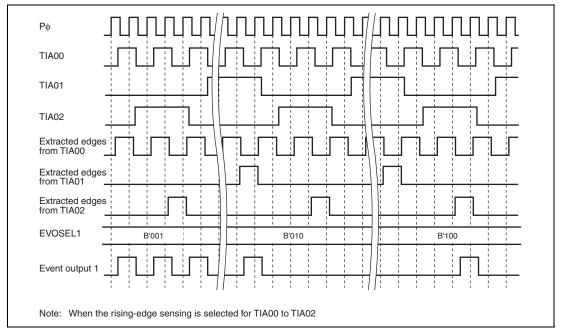


Figure 13.11 Example of Event Output

### 13.11.4 DMA Transfer

The A-DMAC can be activated by an input capture interrupt request. DAM transfer by the A-DMAC transfers data between the input capture register and on-chip RAM. Six 32-bit buffer registers are in the A-DMAC and store data read through DMA transfer. Values previously and currently captured from the input capture register are transferred to the on-chip RAM every interrupt request.

#### Overview of Timer B 13.12

Timer B generates a multiplied-and-corrected clock signal based on an external-event input and supplies the generated signal to other timers. Timer B consists of an edge-interval measuring block, frequency-multiplied clock generator, and frequency-multiplied clock signal corrector.

#### **(1) Edge-Interval Measuring Block**

The edge-interval measuring block measures the intervals between edges of external-event signals input via timer A.

Interrupt requests can be issued for the CPU in response to matches between edge interval measuring counter B0 (TCNTB0) and output compare register B0 (OCRB0). TCNTB0 can also be captured in ICRB2 via ICRB1 on matches between event counter B1 (TCNTB1) and output compare register B1 (OCRB1). Counting by TCNTB1 is driven by the external-event input. This provides a way to measure the intervals between multiple event inputs. Although TCNTB0 is cleared every event input, ICRB1 keeps a running total of the TCNTB0 value. ICRB2 latches the running totals on compare matches of the event counter.

#### **(2)** Frequency-Multiplied Clock Generator

The frequency-multiplied clock generator generates a clock signal by producing from 1 to 4095 cycles in response to an external-event input signal.

A down counter is decremented from values captured from edge-interval measuring counter B0 (TCNTB0) in the edge-interval measuring block. The step size for decrementation is the frequency-multiplication ratio. When the down counter underflows, the multiplied clock (AGCK1) signal is asserted. Note that the clock signal generated by this block is only used within timer B and is not output beyond timer B.

An interrupt request for the CPU can be generated on matches between TCNTB6 and OCRB6. TCNTB6 counts cycles of the frequency-multiplied clock signal.

## (3) Frequency-Multiplied Clock Signal Corrector

The multiplied clock signal (AGCK1) needs to be corrected when two consecutive edge intervals differ significantly, since the earlier interval is referred to in calculating the multiplier for the current interval. The frequency-multiplied clock signal corrector generates a multiplied-and-corrected clock signal (AGCKM) by using three correcting counters (TCNTB3 to TCNTB5) and correcting counter clearing register (TCCLRB). Output of the clock signal thus produced on clock-bus line 5 can be selected by using the clock-bus control register (CBCNT). Other timers can then use this clock as a source for counting.

## 13.12.1 Block Diagram of Timer B

Figure 13.12 is a block diagram of timer B.

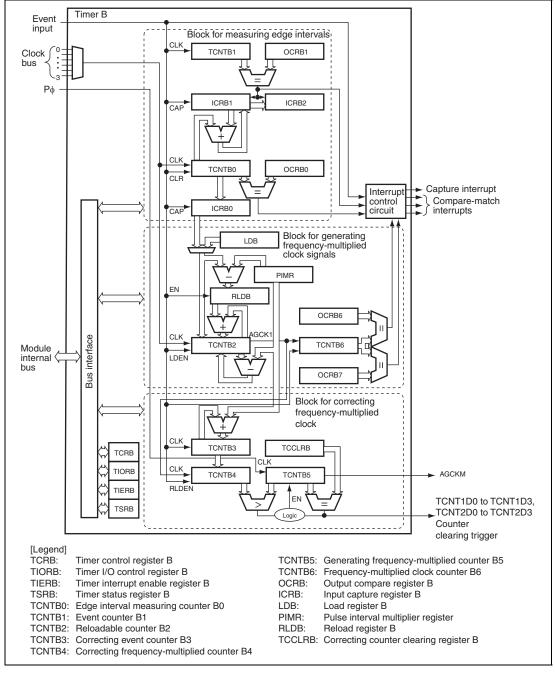


Figure 13.12 Block Diagram of Timer B

# 13.13 Descriptions of Timer B Registers

# 13.13.1 Timer Control Register B (TCRB)

TCRB is an 8-bit readable/writable register that selects the clock driving edge interval measuring counter B0 (TCNTB0) and reloadable counter B2 (TCNTB2).

TCRB can be read from and written to in byte or word units.

TCRB is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1   | 0   |  |
|----------------|---|---|---|---|---|---|-----|-----|--|
|                | - | - | - | - | - | - | CKS | ELB |  |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   |  |
| R/W:           | R | R | R | R | R | R | R/W | R/W |  |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 2 | _        | All 0            | R   | Reserved   |
| =      |          | 7 <b>C</b>       |     | These bits are always read as 0. The write value should always be 0.   |
| 1, 0   | CKSELB   | 00               | R/W | Clock Select B   |
|        |          |                  |     | These bits select the clock driving TCNTB0 and TCNTB2 from clock-bus lines 0 to 3, which are clock signals divided by prescalers 0 to 3. The counters are incremented on the rising edge of the selected clock. To select the clock, stop timer B operation. |
|        |          |                  |     | 00: Clock-bus line 0 is selected   |
|        |          |                  |     | 01: Clock-bus line 1 is selected   |
|        |          |                  |     | 10: Clock-bus line 2 is selected   |
|        |          |                  |     | 11: Clock-bus line 3 is selected   |

# 13.13.2 Timer I/O Control Register B (TIORB)

TIORB is an 8-bit readable/writable register that selects the source of the frequency-multiplied clock and enables and disables the externally input signals, loading data, and correcting frequency-multiplied clock. TIORB also controls multiplied-and-corrected clock generating counter B5 (TCNTB5) and output compare register B6 (OCRB6).

TIORB can be read from and written to in byte or word units.

TIORB is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7         | 6           | 5          | 4    | 3   | 2 | 1 | 0    |
|----------------|-----------|-------------|------------|------|-----|---|---|------|
|                | LD<br>SEL | CTC<br>NTB5 | EVC<br>NTB | LDEN | ccs | 1 | 1 | IOB6 |
| Initial value: | 0         | 0           | 0          | 0    | 0   | 0 | 0 | 0    |
| R/W:           | R/W       | R/W         | R/W        | R/W  | R/W | R | R | R/W  |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 7   | LDSEL    | 0                | R/W | Loading Data Select  |
|     |          |                  |     | Selects the register to be loaded to reloadable counter B2 (TCNTB2) and to be used for calculating data to be loaded to reload register B (RLDB) from ICRB0 or LDB.  |
|     |          |                  |     | 0: ICRB0 is selected   |
|     |          |                  |     | 1: LDB is selected   |
| 6   | CTCNTB5  | 0                | R/W | Count Control B5   |
|     |          |                  |     | Selects whether or not the multiplied-and-corrected clock generating counter B5 (TCNTB5) is stopped.   |
|     |          |                  |     | Setting this bit to 1 stops TCNTB5 and the multiplied-and-corrected clock to be output to other timers. Even if the counter is stopped, it is not cleared. Clearing this bit to 0 resumes TCNTB5 and multiplied-and-corrected clock. |
|     |          |                  |     | 0: TCNTB5 is in operation  |
|     |          |                  |     | 1: TCNTB5 is stopped   |

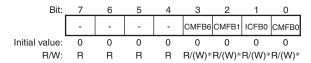
| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 5    | EVCNTB   | 0                | R/W | Event Control B   |
|      |          |                  |     | Disables and enables the externally input events. Clearing this bit to 0 disables the input. Setting this bit to 1 enables the input with which input capture or generating the multiplied-and-corrected clock signal.  |
|      |          |                  |     | 0: Disables the externally input events   |
|      |          |                  |     | 1: Enables the externally input events  |
| 4    | LDEN     | 0                | R/W | Load Enable   |
|      |          |                  |     | Selects whether or not the values in reloadable counter B2 (TCNTB2) and reload register B (RLDB) are updated on input capture by ICRB0.   |
|      |          |                  |     | 0: TCNTB2 and RLDB are updated on ICEB0 input capture   |
|      |          |                  |     | TCNTB2 and RLDB are not updated on ICEB0 input capture  |
| 3    | CCS      | 0                | R/W | Counter Correction Select   |
|      |          |                  |     | Selects whether or not multiplied-and-corrected clock counter B4 (TCNTB4) is stopped when TCNTB3 = TCNTB4.  |
|      |          |                  |     | 0: TCNTB4 is in operation when TCNTB3 = TCNTB4  |
|      |          |                  |     | 1: TCNTB4 is stopped when TCNTB3 = TCNTB4   |
| 2, 1 | _        | All 0            | R   | Reserved  |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 0    | IOB6     | 0                | R/W | I/O Control B6  |
|      |          |                  |     | Enables and disables compare match between TCNTB6 and OCRB6. When this bit is cleared to 0, compare match between TCNTB6 and OCRB6 is disabled. When it is set to 1, compare match between TCNTB6 and OCRB6 is enabled. |
|      |          |                  |     | When bit CMEB6 in timer interrupt enable register B (TIERB) is set to 1, a compare-match interrupt request is issued for the CPU.   |
|      |          |                  |     | 0: Compare match between TCNTB6 and OCRB6 is disabled   |
|      |          |                  |     | 1: Compare match between TCNTB6 and OCRB6 is enabled  |

## 13.13.3 Timer Status Register B (TSRB)

TSRB is an 8-bit readable/writable register indicating that input capture and compare match has occurred. When interrupts are enabled by the corresponding bits in timer interrupt enable register B (TIERB), an interrupt request is issued for the CPU.

TSRB can be read from and written to in byte or word units.

TSRB is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Note: \* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

|        |          | 1.22.1           |        |   |
|--------|----------|------------------|--------|---|
| Bit    | Bit Name | Initial<br>Value | R/W    | Description   |
| 7 to 4 | _        | All 0            | R      | Reserved  |
|        |          |                  |        | These bits are always read as 0. The write value should always be 0.  |
| 3      | CMFB6    | 0                | R/(W)* | Compare Match Flag B6   |
|        |          |                  |        | Indicates that compare match between output compare register B6 (OCRB6) and TCNTB6 has occurred. When this bit is read as 1, the compare match has occurred.  |
|        |          |                  |        | This bit cannot be set to 1 by software.  |
|        |          |                  |        | To clear this bit, write 0 to this bit after read it as 1. Otherwise, writing 1 to this bit is ignored. Only when the IREG bits in TIERB are set to B'10, this bit is automatically cleared to 0 on compare match between TCNTB6 and OCRB6. |
|        |          |                  |        | 0: No compare match has occurred  |
|        |          |                  |        | [Clearing conditions]   |
|        |          |                  |        | • When writing 0 to this bit after reading it as 1  |
|        |          |                  |        | • When compare match between TCNTB6 and OCRB7 occurs while IREG = B'10  |
|        |          |                  |        | 1: Compare match has occurred   |
|        |          |                  |        | [Setting condition]   |
|        |          |                  |        | When the values in TCNTB6 and OCRB6 match   |

| Bit | Bit Name | Initial<br>Value | R/W    | Description  |
|-----|----------|------------------|--------|--|
| 2   | CMFB1    | 0                | R/(W)* | Compare Match Flag B1  |
|     |          |                  |        | Indicates that compare match between output compare register B1 (OCRB6) and TCNTB1 has occurred. When this bit is read as 1, the compare match has occurred. |
|     |          |                  |        | This bit cannot be set to 1 by software.   |
|     |          |                  |        | To clear this bit, write 0 to this bit after read it as 1. Otherwise, writing 1 to this bit is ignored.  |
|     |          |                  |        | 0: No compare match has occurred   |
|     |          |                  |        | [Clearing condition]   |
|     |          |                  |        | When writing 0 to this bit after reading it as 1   |
|     |          |                  |        | 1: Compare match has occurred  |
|     |          |                  |        | [Setting condition]  |
|     |          |                  |        | When the values in TCNTB1 and OCRB1 match  |
| 1   | ICFB0    | 0                | R/(W)* | Input Capture Flag B0  |
|     |          |                  |        | Indicates that input capture by input capture registerB0 (ICRB0) has occurred. When this bit is read as 1, the input capture has occurred.                   |
|     |          |                  |        | This bit cannot be set to 1 by software.   |
|     |          |                  |        | To clear this bit, write 0 to this bit after read it as 1. Otherwise, writing 1 to this bit is ignored.  |
|     |          |                  |        | 0: No input capture has occurred   |
|     |          |                  |        | [Clearing condition]   |
|     |          |                  |        | • When writing 0 to this bit after reading it as 1   |
|     |          |                  |        | 1: Input capture has occurred  |
|     |          |                  |        | [Setting condition]  |
|     |          |                  |        | When the value in TCNTB0 is loaded to ICRB0 by an input capture event as a trigger   |

| Bit | Bit Name | Initial<br>Value | R/W    | Description  |
|-----|----------|------------------|--------|--|
| 0   | CMFB0    | 0                | R/(W)* | Compare Match Flag B0  |
|     |          |                  |        | Indicates that compare match between output compare register B0 (OCRB0) and TCNTB0 has occurred. When this bit is read as 1, the compare match has occurred. |
|     |          |                  |        | This bit cannot be set to 1 by software.   |
|     |          |                  |        | To clear this bit, write 0 to this bit after read it as 1. Otherwise, writing 1 to this bit is ignored.  |
|     |          |                  |        | 0: No compare match has occurred   |
|     |          |                  |        | [Clearing condition]   |
|     |          |                  |        | When writing 0 to this bit after reading it as 1   |
|     |          |                  |        | 1: Compare match has occurred  |
|     |          |                  |        | [Setting condition]  |
|     |          |                  |        | When the values in TCNTB0 and OCRB0 match  |

Note: \* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

## 13.13.4 Timer Interrupt Enable Register B (TIERB)

TIERB is an 8-bit readable/writable register that enables and disables interrupt requests on input capture and compare match.

TIERB can be read from and written to in byte or word units.

TIERB is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5   | 4   | 3     | 2     | 1     | 0     |
|----------------|---|---|-----|-----|-------|-------|-------|-------|
|                | - | - | IRI | EG  | СМЕВ6 | CMEB1 | ICEB0 | СМЕВ0 |
| Initial value: | 0 | 0 | 0   | 0   | 0     | 0     | 0     | 0     |
| R/W:           | R | R | R/W | R/W | R/W   | R/W   | R/W   | R/W   |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 7, 6 | _        | All 0            | R   | Reserved   |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit  | Bit Name  | Initial<br>Value | R/W | Description   |
|------|-----------|------------------|-----|---|
| 5, 4 | IREG[1:0] | 00               | R/W | Interrupt Enable Edge   |
|      |           |                  |     | These bits select the timing at which an interrupt request by CMFB6 in TSRB is output.  |
|      |           |                  |     | 00: An interrupt request is output when CMFB6 is enabled  |
|      |           |                  |     | 01: An interrupt request is output when the first externally input event is detected after CMFB6 is enabled   |
|      |           |                  |     | 10: An interrupt request is output when the second<br>externally input event is detected after CMFB6 is<br>enabled. However, if compare match B7 has occurred<br>before the second event is detected, the interrupt<br>request is not output. |
|      |           |                  |     | 11: Setting prohibited  |
| 3    | CMEB6     | 0                | R/W | Compare Match Interrupt B6 Enable   |
|      |           |                  |     | Enables and disables the output of the interrupt request when compare match flag B6 (CMFB6) is set to 1. The timing at which the interrupt request is output is set by the IREG bits.   |
|      |           |                  |     | 0: Output of compare match interrupt B6 is disabled   |
|      |           |                  |     | 1: Output of compare match interrupt B6 is enabled  |
| 2    | CMEB1     | 0                | R/W | Compare Match Interrupt B1 Enable   |
|      |           |                  |     | Enables and disables the output of the interrupt request when compare match flag B1 (CMFB1) is set to 1.  |
|      |           |                  |     | 0: Output of compare match interrupt B1 is disabled   |
|      |           |                  |     | 1: Output of compare match interrupt B1 is enabled  |
| 1    | ICEB0     | 0                | R/W | Input Capture Interrupt B0 Enable   |
|      |           |                  |     | Enables and disables the output of the interrupt request when input capture flag B0 (ICFB0) is set to 1.  |
|      |           |                  |     | 0: Input capture flag B0 (ICFB0) is disabled  |
|      |           |                  |     | 1: Input capture flag B0 (ICFB0) is enabled   |
| 0    | CMEB0     | 0                | R/W | Compare Match Interrupt B0 Enable   |
|      |           |                  |     | Enables and disables the output of the interrupt request when compare match flag B0 (CMFB0) is set to 1.  |
|      |           |                  |     | 0: Output of compare match interrupt B0 is disabled   |
|      |           |                  |     | 1: Output of compare match interrupt B0 is enabled  |

### 13.13.5 Edge Interval Measuring Counter B0 (TCNTB0)

TCNTB0 is a 32-bit readable/writable register that functions as a counter driven by the clock selected in the clock select B bits of timer control register B (TCRB). TCNTB0 is cleared to H'00000001 on input capture by an externally input event.

TCNTB0 is started when the timer B enable bit (TBE) in ATU-III master enable register (ATUENR) is set to 1. Clearing the TBE bit to 0 stops the counting but the counter value is not cleared.

TCNTB0 can be read from and written to in longword units.

TCNTB0 is initialized to H'0000 0001 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     | CN. | TB0 |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     | CN. | TB0 |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   |
| R/W:           | R/W |

| Bit     | Bit Name | Initial Value | R/W | Description                            |
|---------|----------|---------------|-----|--|
| 31 to 0 | CNTB0    | H'00000001    | R/W | Edge Interval Count                    |
|         |          |               |     | These bits store 32-bit counter value. |

## 13.13.6 Input Capture Register B0 (ICRB0)

ICRB0 is a 32-bit read-only register that is loaded with the value in TCNTB0 when an externally input event is detected. At this time, bit ICFB0 in timer status register B (TSRB) is set to 1.

On this input capture of ICRB0, TCNTB0 is cleared to H'00000001.

ICRB0 can be read from in longword units.

ICRB0 is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|                |    |    |    |    |    |    |    | IC | В0 |    |    |    |    |    |    |    |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |
| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|                |    |    |    |    |    |    |    | IC | В0 |    |    |    |    |    |    |    |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

| Bit     | Bit Name | Initial<br>Value | R/W | Description                             |
|---------|----------|------------------|-----|---|
| 31 to 0 | ICB0     | All 0            | R   | Input Capture B0                        |
|         |          |                  |     | These bits store 32-bit captured value. |

## 13.13.7 Output Compare Register B0 (OCRB0)

OCRB0 is a 32-bit readable/writable register that is constantly compared with free-running counter B0 (TCNTB0). When they match, bit CMFB0 in timer status register B (TSRB) is set to 1. An interrupt request is issued for the CPU by setting bit CMEB0 in timer interrupt enable register B (TIERB) to 1.

OCRB0 can be read from and written to in longword units.

OCRB0 is initialized to H'FFFFFFF by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     | 00  | B0  |     |     |     |     |     |     |     |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     | 00  | B0  |     |     |     |     |     |     |     |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W:           | R/W |

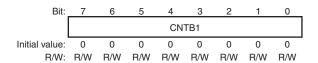
| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 31 to 0 | OCB0     | All 1            | R/W | Output Compare B0  |
|         |          |                  |     | These bits store 32-bit data to be compared with TCNTB0. |

### 13.13.8 Event Counter B1 (TCNTB1)

TCNTB1 is an 8-bit readable/writable register that counts the externally input events when the TBE bit in ATU-III master enable register (ATUENR) is set to 1.

TCNTB1 can be read from and written to in byte or word units.

TCNTB1 is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



| Bit    | Bit Name | Initial<br>Value | R/W | Description                           |
|--------|----------|------------------|-----|---------------------------------------|
| 7 to 0 | CNTB1    | All 0            | R/W | Event Count B1                        |
|        |          |                  |     | These bits store 8-bit counter value. |

### 13.13.9 Output Compare Register B1 (OCRB1)

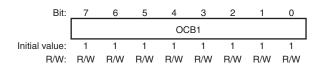
OCRB1 is an 8-bit readable/writable register that is constantly compared with event counter B1 (TCNTB1). Bit CMFB1 in timer status register B (TSRB) is set to 1 on the first edge of the Pφ clock after the values in this register and TCNTB1.

An interrupt request is issued for the CPU on compare match by setting bit CMEB1 in timer interrupt enable register B (TIERB).

On this compare match, the value in input capture registerB1 (ICRB1) is loaded to input capture registerB2 (ICRB2) and ICRB1 is cleared.

OCRB1 can be read from and written to in byte or word units.

OCRB1 is initialized to H'FF by a power-on reset or a transition to the hardware standby mode.



| Bit    | Bit Name | Initial<br>Value | R/W | Description                                 |
|--------|----------|------------------|-----|---|
| 7 to 0 | OCB1     | All 1            | R/W | Output Compare B1                           |
|        |          |                  |     | These bits store 8-bit data to be compared. |

## 13.13.10 Input Capture Register B1 (ICRB1)

ICRB1 is a 32-bit read-only register. ICRB1 keeps a running total of the value in edge interval measuring counter B0 (TCNTB0) when the externally input event is detected. ICRB1 is cleared on compare match between event counter B1 (TCNTB1) and output compare register B1 (OCRB1).

ICRB1 can be read from in longword units.

ICRB1 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31 | 30   | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|                |    |      |    |    |    |    |    | IC | B1 |    |    |    |    |    |    |    |
| Initial value: | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R    | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |
| Bit:           | 15 | 14   | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|                |    | ICB1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Initial value: | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R    | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

| Bit     | Bit Name | Initial<br>Value | R/W | Description                                  |
|---------|----------|------------------|-----|--|
| 31 to 0 | ICB1     | All 0            | R   | Input Capture B1                             |
|         |          |                  |     | These bits store 32-bit input capture value. |

# 13.13.11 Input Capture Register B2 (ICRB2)

ICRB1 is a 32-bit read-only register. ICRB1 keeps a running total of the value in input capture registerB1 (ICRB1) on compare match between event counter B1 (TCNTB1) and output compare register B1 (OCRB1).

ICRB2 can be read from in longword units.

ICRB2 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31 | 30   | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|                |    |      |    |    |    |    |    | IC | B2 |    |    |    |    |    |    |    |
| Initial value: | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R    | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |
| Bit:           | 15 | 14   | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|                |    | ICB2 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Initial value: | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R    | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

| Bit     | Bit Name | Initial<br>Value | R/W | Description                                  |
|---------|----------|------------------|-----|--|
| 31 to 0 | ICB2     | All 0            | R   | Input Capture B2                             |
|         |          |                  |     | These bits store 32-bit input capture value. |

## 13.13.12 Load Register B (LDB)

LDB is a 32-bit readable/writable register that is aligned with a longword boundary. The lower 24 bits are available.

When the LDSEL bit in timer I/O control register B (TIORB) is set to 1, the value in this register is loaded to reloadable counter B2 (TCNTB2) and reload register B (RLDB).

LDB can be read from and written to in longword units.

LDB is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23          | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-------------|-----|-----|-----|-----|-----|-----|-----|
|                | -   | 1   | ı   | 1   | -   | -   | -   | -   |             |     |     | LD\ | /AL |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0           | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R   | R   | R   | R   | R   | R   | R   | R   | R/W         | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7           | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     | LD\ | <b>V</b> AL |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0           | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W         | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 31 to 24 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 23 to 0  | LDVAL    | All 0            | R/W | Load Value   |
|          |          |                  |     | These bits store 24-bit data to be loaded to TCNTB2 and RLDB.        |

#### 13.13.13 Reload Register B (RLDB)

RLDB is a 32-bit readable/writable register that is aligned with a longword boundary. The upper 24 bits are available.

This register is updated when the externally input event is detected while the LDEN bit in timer I/O control register B (TIORB) is cleared to 0.

The value in input capture registerB0 (ICRB0) or load register B (LDB) minus the value in the PIMR bits in PIM is used for updating. ICRB0 or LDB is set by the LDSEL bit in TIORB. For subtraction on ICRB0 and PIMR, the lower 24 bits of ICRB0 and PIM which is zero-extended.

The value in this register is added to the value in TCNTB2 on the first counter clock after the value in bits reload count B2 (CNTB2) is equal to or less than the value in the pulse interval multiplier register (PIM).

RLDB can be read from and written to in longword units.

RLDB is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     | RLD | VAL |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     | RLD | VAL |     |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

|         |          | Initial |     |  |
|---------|----------|---------|-----|--|
| Bit     | Bit Name | Value   | R/W | Description  |
| 31 to 8 | RLDVAL   | All 0   | R/W | Reload Value   |
|         |          |         |     | These bits store 24-bit reload value.                                |
| 7 to 0  | _        | All 0   | R   | Reserved   |
|         |          |         |     | These bits are always read as 0. The write value should always be 0. |

### 13.13.14 Reloadable Counter B2 (TCNTB2)

TCNTB2 is a 32-bit readable/writable register that functions as a counter driven by the clock selected in bit clock select B (CKSELB) in timer control register B (TCRB). Each decrementation is by the value set in the pulse interval multiplier register (PIMR).

When the TBE bit in ATU-III master enable register (ATUENR) is set to 1, this counter is decremented. Even if the TBE bit is cleared to 0, this counter is not cleared.

This counter is updated when the externally input event is detected while the LEDN bit in timer I/O control register B (TIORB) is cleared to 0.

The value in input capture registerB0 (ICRB0) or load register B (LDB) is used for updating. ICRB0 or LDB is set by the LDSEL bit in TIORB.

The value in reload register B (RLDB) is added to the value stored in this counter on the first counter clock after the value in this counter is equal to or less than the value in PIM. A single pulse whose width is equal to the cycle of the  $P\phi$  clock on reloading. The pulse is the frequency-multiplied clock (AGCK1).

TCNTB2 can be read from and written to in longword units.

TCNTB2 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28              | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |                 |     |     |     | CN. | ГВ2 |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W | R/W | R/W             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit:           | 15  | 14  | 13  | 12              | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     | CN <sup>-</sup> | TB2 |     |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0   | 0   | 0               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W·           | R/W | R/W | R/W | R/W             | R/W | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

|         | Bit   | Initial |     |  |
|---------|-------|---------|-----|--|
| Bit     | Name  | Value   | R/W | Description  |
| 31 to 8 | CNTB2 | All 0   | R/W | Reload Count B2  |
|         |       |         |     | These bits store 24-bit reload counter value.                        |
| 7 to 0  | _     | All 0   | R   | Reserved   |
|         |       |         |     | These bits are always read as 0. The write value should always be 0. |

### 13.13.15 Pulse Interval Multiplier Register (PIMR)

PIMR is a 16-bit readable/writable register that sets the multiplication ratio of the externally input event for generation of the frequency-multiplied clock.

The settable value ranges from 1 (H'001) to 4095 (H'FFF). Do not set the PIM bits to H'000. If the PIM bits are set to H'000, operation cannot be guaranteed.

The value in this register is used in various registers; the step size in decrementation of reloadable counter B2 (TCNTB2); calculation of the value to be input to reload register B (RLDB); calculation of the value to be input to corrected event counter B3 (TCNTB3).

PIMR can be read from and written to in word units.

PIMR is initialized to H'0001 by a power-on reset or a transition to the hardware standby mode.

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

| Bit      | Bit Name | Initial<br>Value | R/W | Description   |
|----------|----------|------------------|-----|---|
| 15 to 12 | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 11 to 0  | PIM      | H'001            | R/W | Pulse Interval Multiplier   |
|          |          |                  |     | These bits set the multiplication ratio for the frequency-multiplied clock. The settable value ranges from 1 to 4095. |

# 13.13.16 Multiplied Clock Counter B6 (TCNTB6)

TCNTB6 is a 32-bit readable/writable register that functions as an up-counter driven by the frequency-multiplied clock (AGCK1). This counter is cleared to H'00000000 when the external input event is detected.

When the TBE bit in ATU-III master enable register (ATUENR) is set to 1, this counter is incremented. Even if the TBE bit is cleared to 0 and counting up is stopped, the counter value is not cleared.

TCNTB6 can be read from and written to in longword units.

TCNTB6 is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     | CN. | TB6 |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     | CN. | TB6 |     | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   |

|          |          | Initial |     |   |
|----------|----------|---------|-----|---|
| Bit      | Bit Name | Value   | R/W | Description   |
| 31 to 12 | CNTB6    | All 0   | R/W | Frequency-Multiplied Clock Count B6   |
|          |          |         |     | These bit store 20-bit counter value driven by the frequency-multiplied clock |
| 11 to 0  | _        | All 0   | R   | Reserved  |
|          |          |         |     | These bits are always read as 0. The write value should always be 0.          |

## 13.13.17 Output Compare Register B6 (OCRB6)

OCRB6 is a 32-bit readable/writable register that is aligned with a longword boundary. The upper 20 bits are available.

This register enables and disables compare match between multiplied clock counter B6 (TCNTB6) and OCRB6 when bit IOB6 in timer I/O control register B (TIORB) is set.

When the edge of the frequency-multiplied clock (AGCK1) is input while the values in this counter and TCNTB6 have matched with compare match enabled by bit IOB6, bit CMFB6 in timer status register B (TSRB) is set to 1. An interrupt request is issued for the CPU by setting bit CMEB6 in timer interrupt enable register B (TIERB) to 1.

OCRB6 can be read from and written to in longword units.

OCRB6 is initialized to H'FFFFF000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     | OC  | B6  |     |     |     |     |     |     |     |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     | OC  | B6  |     | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 1   | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   |

|          |          | Initial |     |  |
|----------|----------|---------|-----|--|
| Bit      | Bit Name | Value   | R/W | Description  |
| 31 to 12 | OCB6     | All 1   | R/W | Output Compare B6  |
|          |          |         |     | These bits store 20-bit data to be compared.                         |
| 11 to 0  | _        | All 0   | R   | Reserved   |
|          |          |         |     | These bits are always read as 0. The write value should always be 0. |

## 13.13.18 Output Compare Register B7 (OCRB7)

OCRB7 is a 32-bit readable/writable register that is aligned with a longword boundary. The upper 20 bits are available.

When the edge of the frequency-multiplied clock (AGCK1) is input while the values in this counter and TCNTB6 have matched with the IREG bits in timer interrupt enable register B (TIERB) set to B'10, bit CMFB6 in timer status register B (TSRB) is cleared to 0.

There is no status flag indicating the compare match. An interrupt request cannot be issued for the CPU.

OCRB7 can be read from and written to in longword units.

OCRB7 is initialized to H'FFFFF000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     | OC  | B7  |     |     |     |     |     |     |     |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     | OC  | B7  |     | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 1   | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   |

|          |          | Initial |     |  |
|----------|----------|---------|-----|--|
| Bit      | Bit Name | Value   | R/W | Description  |
| 31 to 12 | OCB7     | All 1   | R/W | Output Compare B7  |
|          |          |         |     | These bit store 20-bit data to be compared.                          |
| 11 to 0  | _        | All 0   | R   | Reserved   |
|          |          |         |     | These bits are always read as 0. The write value should always be 0. |

## 13.13.19 Correcting Event Counter B3 (TCNTB3)

TCNTB3 is a 32-bit readable/writable register.

When the externally input event is detected, the value in this counter is transferred to multipliedand-corrected clock counter B4 (TCNTB4) and then is incremented by the value in the pulse interval multiplier register (PIMR)

When the TBE bit in ATU-III master enable register (ATUENR) is set to 1, this counter is incremented. Even if the TBE bit is cleared to 0, the counter value is not cleared.

TCNTB3 can be read from and written to in longword units.

TCNTB3 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     | CN. | TB3 |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     | CN. | ТВ3 |     | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   |

| Bit      | Bit Name | Initial<br>Value | R/W | Description   |
|----------|----------|------------------|-----|---|
| 31 to 12 | CNTB3    | All 0            | R/W | Correcting Event Count B3   |
|          |          |                  |     | These bits store 20-bit event count value.                        |
| 11 to 0  | _        | All 0            | R   | Reserved  |
|          |          |                  |     | This bit is always read as 0. The write value should always be 0. |

# 13.13.20 Multiplied-and-Corrected Clock Counter B4 (TCNTB4)

TCNTB4 is a 32-bit readable/writable register that is a 20-bit up-counter. The value in TCNTB3 is loaded to this counter when the externally input event is detected.

This up-counter is driven by the frequency-multiplied clock (AGCK1) output by reloadable counter B2 (TCNTB2) and is cleared to H'00000 when the externally input event is detected while CNTB3 = H'00000.

When the values in this counter and TCNTB3 match with the CCS bit in timer I/O control register B (TIORB) set to 1, counting is stopped.

When the TBE bit in ATU-III master enable register (ATUENR) is set to 1, this counter is incremented. Even if the TBE bit is cleared to 0 and counting is stopped, the counter value is not cleared.

TCNTB4 can be read from and written to in longword units.

TCNTB4 is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30              | 29  | 28  | 27  | 26  | 25  | 24              | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----------------|-----|-----|-----|-----|-----|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |                 |     |     |     |     |     | CN <sup>-</sup> | TB4 |     |     |     |     |     |     |     |
| Initial value: | 0   | 0               | 0   | 0   | 0   | 0   | 0   | 0               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W             | R/W | R/W | R/W | R/W | R/W | R/W             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit:           | 15  | 14              | 13  | 12  | 11  | 10  | 9   | 8               | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     | CN <sup>-</sup> | ТВ4 |     | -   | -   | -   | -               | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0               | 0   | 0   | 0   | 0   | 0   | 0               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W             | R/W | R/W | R   | R   | R   | R               | R   | R   | R   | R   | R   | R   | R   | R   |

|          |          | Initial |     |  |
|----------|----------|---------|-----|--|
| Bit      | Bit Name | Value   | R/W | Description  |
| 31 to 12 | CNTB4    | All 0   | R/W | Multiplied-and-Corrected Clock Count B4                              |
|          |          |         |     | These bits store 20-bit frequency-multiplied clock count value.      |
| 11 to 0  | _        | All 0   | R   | Reserved   |
|          |          |         |     | These bits are always read as 0. The write value should always be 0. |

#### 13.13.21 Multiplied-and-Corrected Clock Generating Counter B5 (TCNTB5)

TCNTB5 is a 32-bit readable/writable register.

This counter is enabled by the TBE bit in ATU-III master enable register (ATUENR) and the count control B5 bit (CTCNTB5). Incrementation of this counter is driven by the P $\phi$  clock as long as the counter value is less than the value in multiplied-and-corrected clock counter B4 (TCNTB4). Incrementation is stopped when the counter value matches the value in correcting counter clearing register B (TCCLRB).

The value in this counter is corrected and cleared when the externally input event is detected while TCNTB3 = H'00000. The value for the clearing depends on the counter value. When the counter value is equal to the value in TCCLRB, this counter is cleared to H'00001000. When the counter value is not equal to the value in TCCLRB, incrementation continues until the counter value reaches the value in TCCLRB and then TCNTB5 is cleared to H'00001000.

Every incrementation of this counter, a single pulse of the multiplied-and-corrected clock (AGCKM) is output. The clock can be output on clock-bus line 5 by setting the CB5SEL bit in clock bus control register (CBCNT) to 1. The output of the clock is temporarily stopped by altering the setting in bit CTCNTB5.

TCNTB5 can be read from and written to in longword units.

TCNTB5 is initialized to H'0000 1000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30              | 29  | 28  | 27  | 26  | 25  | 24              | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----------------|-----|-----|-----|-----|-----|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |                 |     |     |     |     |     | CN <sup>-</sup> | TB5 |     |     |     |     |     |     |     |
| Initial value: | 0   | 0               | 0   | 0   | 0   | 0   | 0   | 0               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W             | R/W | R/W | R/W | R/W | R/W | R/W             | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Dit.           | 45  | 4.4             | 10  | 10  |     | 10  | _   | 0               | 7   | _   | _   |     | 0   | 0   |     | 0   |
| Bit:           | 15  | 14              | 13  | 12  | 11  | 10  | 9   | 8               | /   | 6   | 5   | 4   | 3   | 2   | - 1 | 0   |
|                |     | CN <sup>-</sup> | ТВ5 |     | 1   | 1   | -   | -               | -   | -   | -   | 1   | -   | 1   | 1   | -   |
| Initial value: | 0   | 0               | 0   | 1   | 0   | 0   | 0   | 0               | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W             | R/W | R/W | R   | R   | R   | R               | R   | R   | R   | R   | R   | R   | R   | R   |

|          |          | Initial |     |  |
|----------|----------|---------|-----|--|
| Bit      | Bit Name | Value   | R/W | Description  |
| 31 to 12 | CNTB5    | H'00001 | R/W | Multiplied-and-Corrected Clock Generation Count B5                   |
|          |          |         |     | These bit store 20-bit multiplied-and-corrected clock count value.   |
| 11 to 0  | _        | All 0   | R   | Reserved   |
|          |          |         |     | These bits are always read as 0. The write value should always be 0. |

## 13.13.22 Correcting Counter Clearing Register B (TCCLRB)

TCCLRB is a 32-bit readable/writable register.

TCCLRB is constantly compared with TCNTB5. When they match, TCNTB5 is stopped and a counter clearing trigger is output to timer D. TCNT1Dn and TCNT2Dn in timer D are separately cleared by setting the corresponding counter clearing enable bit in timer control register Dn (TCRDn).

TCCLRB can be read from and written to in longword units.

TCCLRB is initialized to H'000000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     | CCI | _RB |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     | CCI | _RB |     | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   |

|          |          | Initial |     |  |
|----------|----------|---------|-----|--|
| Bit      | Bit Name | Value   | R/W | Description  |
| 31 to 12 | CCLRB    | All 0   | R/W | Correcting Counter Clear B   |
|          |          |         |     | These bits store 20-bit correcting counter clear value.              |
| 11 to 0  | _        | All 0   | R   | Reserved   |
|          |          |         |     | These bits are always read as 0. The write value should always be 0. |

# 13.14 Operations of Timer B

### 13.14.1 Edge Interval Measuring Function and Edge Input Stopping Function

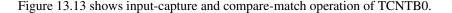
For timer B, input-capture and compare-match operations are unconditionally performed with input capture register B0 (ICRB0) and output compare register B0 (OCRB0), respectively. These registers are connected to free-running counter B0 (TCNTB0).

Operation of timer B is started by setting the TBE bit in the ATU-III master enable register (ATUENR).

ICRB0 captures the TCNTB0 value when an event (the AGCK signal) is input via timer A. After that, TCNTB0 is set to H'00000001. If the interrupt is enabled by the setting in timer interrupt enable register B (TIERB), an interrupt request is also generated for the CPU. This enables measurement of the interval between external event edges.

The value captured by ICRB0 is transferred to the frequency-multiplied clock signal generator, where it is used to be set in reloadable counter B2 (TCNTB2) and reload register B (RLDB).

A compare-match interrupt when the TCNTB0 value reaches the OCRB0 value can also be requested. This interrupt indicates that active edge input has stopped for at least time equivalent to the setting in OCRB0.



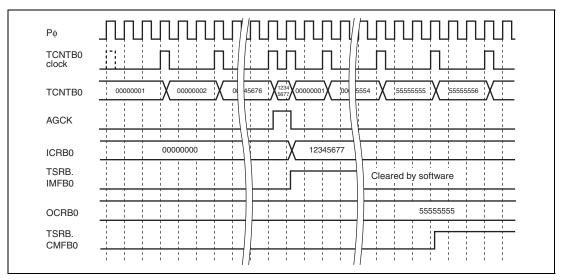


Figure 13.13 Input-Capture and Compare-Match Operation of TCNTB0

Event counter B1 (TCNTB1) counts the input of signals indicating external events (AGCK). When a predetermined value is set in output compare register B1 (OCRB1) and TCNTB1 reaches that value, a compare-match occurs. At this time, input capture register B2 (ICRB2) captures the value in input capture register B1 (ICRB1). An interrupt request for the CPU will also be generated if bit CMEB1 in TIERB is set to 1. The interrupt indicates that the input of edges of the external-event signal has stopped for at least time equivalent to the setting in OCRB1.

The external event signal (AGCK) drives the capturing of TCNTB0 values in ICRB1. Moreover, latching of ICRB1 in ICRB2 on matches between TCNTB1 and OCRB1 can also be selected. This enables the measurement of multiple edge-to-edge intervals.

Figure 13.14 shows compare-match operation of TCNTB1 and capture operation of ICRB1 and ICRB2.

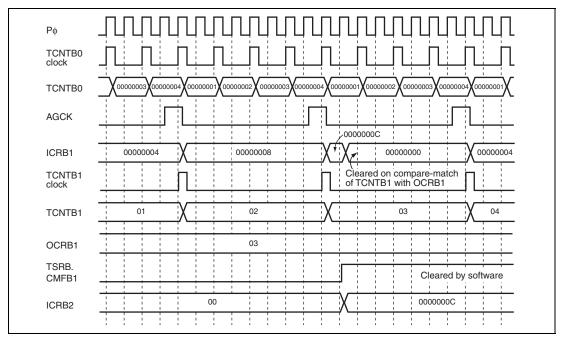


Figure 13.14 Compare-Match Operation of TCNTB1 and Capture Operation of ICRB1 and ICRB2

#### 13.14.2 Frequency-Multiplied Clock Generator

The frequency-multiplied clock generator generates a clock signal (AGCK1) for use within timer B. The cycle of this clock signal is obtained by dividing the intervals between selected transitions of the external-event input (AGCK) by the value in the pulse interval multiplier register (PIMR).

On the selected transition of the external-event input signal, the lower 24 bits of the value captured in ICRB0 of the edge-interval measuring block are transferred to reloadable counter B2 (TCNTB2). At the same time, the value transferred to TCNTB2 minus the value in PIMR (PIM) is transferred to reload register B (RLDB).

When the LDSEL bit in timer I/O control register B (TIORB) is set to 1, the value in load register (LDB) instead of that in ICRBO can be transferred to TCNTB2 and RLDB.

Reloadable counter B2 (TCNTB2) is driven by the clock selected by the CKSELB bits in timer control register B (TCRB). Each decrementation is by the value set in PIMR. When the value in the down counter is less than or equal to that of the PIM bits, RLDB is automatically read out into TCNTB2, which again starts to count down with the same step size (the value in PIMR). A single pulse of the multiplied clock signal (AGCK1) is output in synchronization with the reloading of TCNTB2. The pulse width is equal to the cycle of the  $P\phi$  clock.

Frequency-multiplied clock counter B6 (TCNTB6) is driven by the frequency-multiplied clock signal (AGCK1). Compare match operation can be performed when the values in TCNTB6 and output compare register B6 (OCRB6) match.

Incrementaion of TCNTB6 on the assertion of AGCK1 is unconditional. The values in TCNTB6 and output compare register B6 (OCRB6) are tested for matches, and an interrupt request will be generated for the CPU when the values match, if this interrupt has been enabled in timer interrupt enable register B (TIERB). The IREG bits in TIERB can be set so that the interrupt is generated on the match, on the first AGCK pulse after the match, or on the second AGCK pulse after the match.

Since AGCK1 is generated with reference to the previous edge-to-edge interval, if two consecutive edge intervals differ significantly, the clock will not be generated correctly. To correct this error, AGCK1 is corrected by the frequency-multiplied clock signal corrector which produces the multiplied-and-corrected clock (AGCKM; described in section 13.14.3, Frequency-Multiplied Clock Signal Corrector).

Figures 13.15 and 13.16 show counting operations with reloading and the frequency-multiplied clock and figures 13.17 and 13.18 show the generation of interrupt requests on matches between TCNTB6 and CMFB6, enabled or disabled by the setting in IREG.

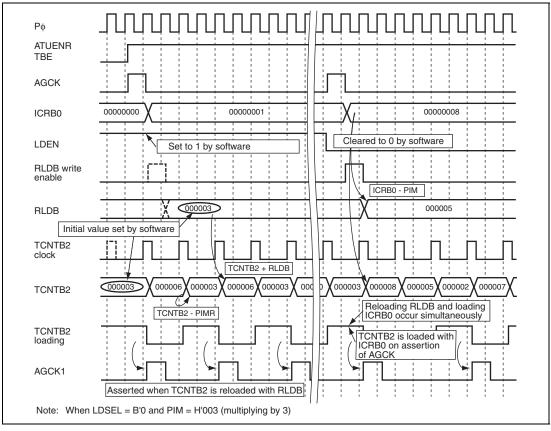


Figure 13.15 Counting Operations with Reloading and Output of Frequency-Multiplied Clock (1)

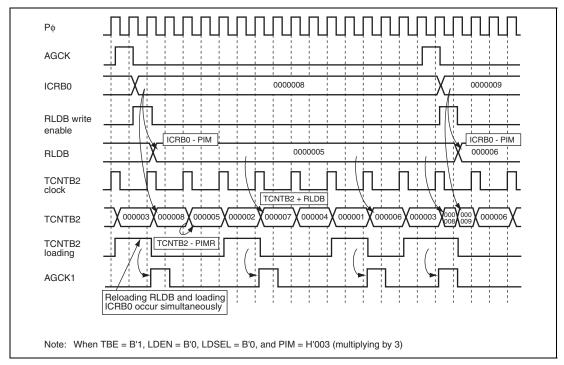


Figure 13.16 Counting Operations with Reloading and **Output of Frequency-Multiplied Clock (2)** 

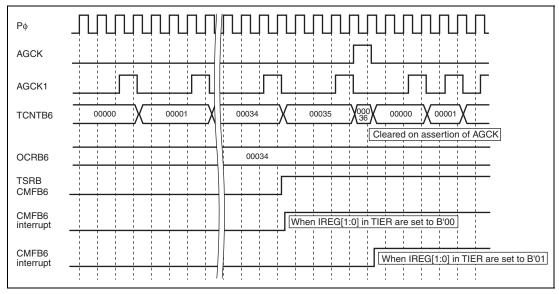


Figure 13.17 Compare-Match Operation of TCNTB6 and Output of CMFB6 Interrupt (IREG = B'00, B'01)

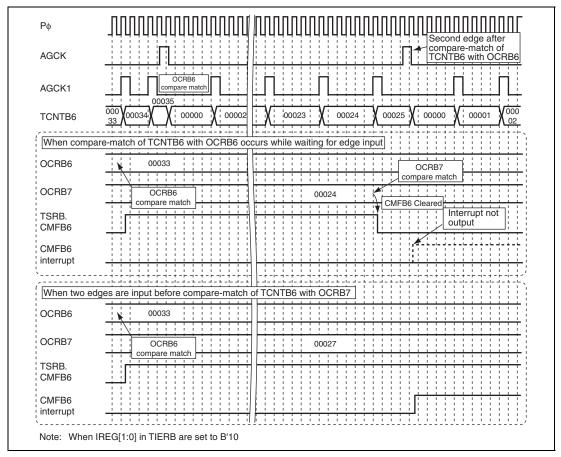


Figure 13.18 CMFB6 Interrupt Output when IREG = B'10

#### 13.14.3 Frequency-Multiplied Clock Signal Corrector

The frequency-multiplied clock signal which is generated by dividing the intervals between external event inputs by the multiplication ratio set in the PIM bits in PIMR can be corrected by using correcting event counter B3 (TCNTB3), multiplied-and-corrected clock counter B4 (TCNTB4), multiplied-and-corrected clock generating counter B5 (TCNTB5), and correcting counter clearing register B (TCCLRB).

TCNTB3 is a 20-bit up-counter that is driven by the external event input (AGCK). On the selected transition of the AGCK signal, the value in TCNTB3 is transferred to TCNTB4, after which TCNTB3 is incremented by the value in the PIM bits.

TCNTB4 is a 20-bit up-counter that is driven by the multiplied clock signal (AGCK1). TCNTB3 is loaded to TCNTB4 with the AGCK input as a trigger, and incrementation of TCNTB4 is driven by the AGCK1 input.

The counter correcting select bit (CCS) in TIORB controls counting by TCNTB4; that is, it selects whether or not counting stops when TCNTB3 = TCNTB4.

TCNTB5 is a 20-bit up-counter that is driven by the  $P\phi$  clock, meaning that it operates at a high speed. TCNTB5 is constantly compared with TCNTB4 and is incremented as long as its value is lower than that in TCNTB4. Each time TCNTB5 is actually incremented, it produces a single pulse whose width is equal to one cycle of the  $P\phi$  clock. pulse of the peripheral clock signal, namely the multiplied-and -corrected clock signal (AGCKM), for which output on clock-bus line 5 can be selected by bit CB5SEL in the clock bus control register (CBCNT). The AGCKM signal is then available on clock-bus line 5 as a source to drive counting by other timers.

As state above, TCNTB5 is not incremented when its value is greater that in TCNTB4 (for example, after TCNTB3 has been loaded to TCNTB4), TCNTB5 can also be disabled by the count control B5 (CTCNTB5) bit in timer I/O control register B (TIORB). This halts the output of the AGCKM signal.

As long as its value is lower that that in TCNTB4, TCNTB5 is incremented until it reaches the value in correcting counter clearing register B (TCCLRB). Incrementation of TCNTB5 then stops, regardless of the relation between its value and that of TCNTB4. In addition, counters of timer D (TCNT1Dn and TCNT2Dn) can be separately cleared by this match as a trigger when the corresponding counter clearing enable bit (C1CEDn/C2CEDn) in timer control register Dn (TCRDn) is set to 1.

TCNTB4 is unconditionally cleared to H'00000000 when a pulse of the external-event signal (AGCK) is input while TCNTB3 = H'00000000. TCNTB5 is unconditionally set to H'00001000 when a pulse of the external-event signal (AGCK) is input while TCNTB3 = H'00000000. However, when TCNTB5 has not reached TCCLRB, TCNTB5 is incremented until it reaches TCCLRB. After that, it is set to H'00001000.

Figure 13.19 shows operations of TCNTB3 and TCNTB4, figure 13.20 shows operation when TCNTB5 is being started up, figure 13.21 shows TCNTB5 operation with correction at the end of a cycle, and figure 13.22 shows TCNTB5 operation with no correction at the end of a cycle.

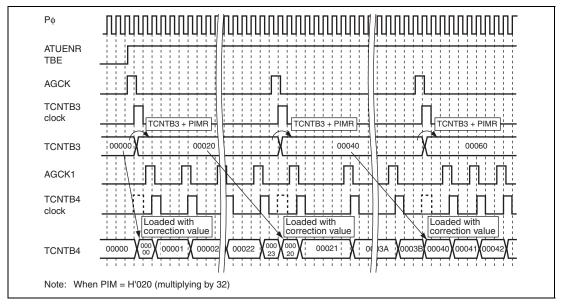


Figure 13.19 Operation of TCNTB3 and TCNTB4

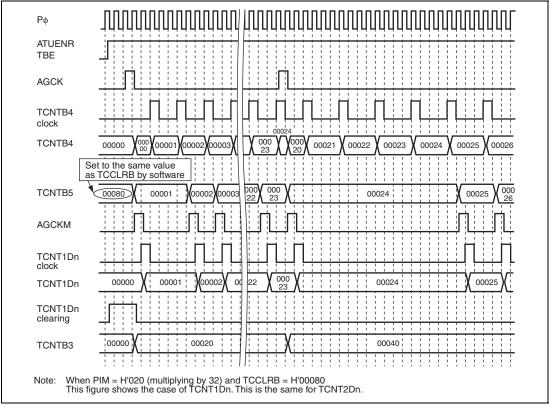


Figure 13.20 TCNTB5 Operation (at Start-Up)

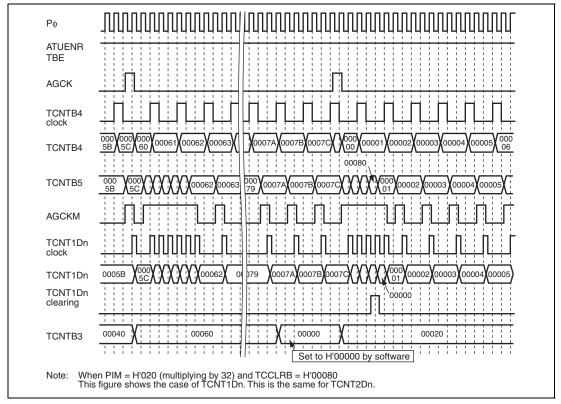


Figure 13.21 Operation of TCNTB5 (with Correction at End of Cycle)

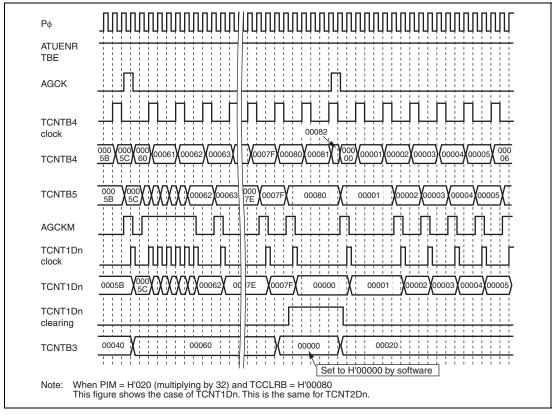


Figure 13.22 Operation of TCNTB5 (without Correction at End of Cycle)

#### 13.15 Overview of Timer C

Timer C consists of five subblocks and has the following functions.

- Input capture and output compare matches
- Choice of rising edge, falling edge, or both edge sensing as the edge of input capture trigger signal
- Output of a waveform on compare match
- Choice of a logical one, a logical zero, or a toggled output by setting a register
- Output of an interrupt request on input capture or compare match
- GRC00, GRC10, GRC20, GRC30, and GRC40, which are used for interrupt requests for the DMAC, can be cleared by the ACK signal.
- Output of an interrupt request on timer counter overflow
- Clearing counter GRCn0 on compare match (not supported by GRCn1 to GRCn3)
- Output of forced compare match by setting the forced compare match bit
- Each input capture trigger input has noise canceling function.

#### 13.15.1 **Block Diagram of Timer C**

Timer C consists of five subblocks. Each subblock consists of timer counter C (TCNTC), four general registers (GRCm), and controller. The general registers can be used for input capture/compare match operations and input capture trigger input and output compare output signals (TIOCnm) are available.

The initial output value on the TIOCnm pin is 0 for output compare operation. During operation, the previous state is reflected.

Figure 13.23 is a block diagram of timer C.

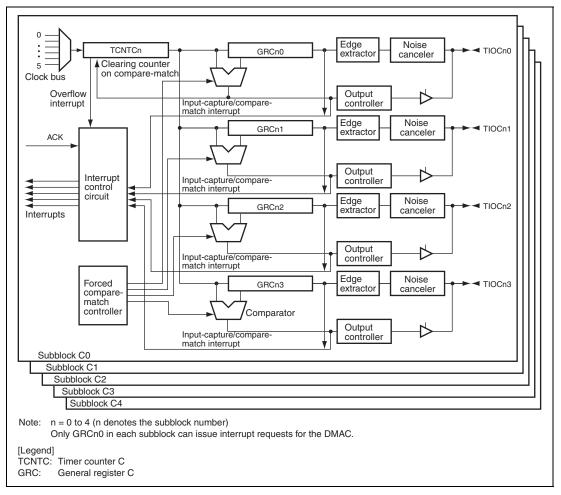


Figure 13.23 Block Diagram of Timer C

#### **Description of Timer C Registers** 13.16

#### **Timer Start Register C (TSTRC)** 13.16.1

TSTRC is an 8-bit readable/writable register that enables and disables timer counter Cn (TCNTCn) in subblocks C0 to C4. When the both the STRC bits in this register and the TCE bit in ATU-III master enable register (ATUENR) are set to 1, counting is enabled.

TSTRC can be read from and written to in byte or word units.

TSTRC is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4     | 3     | 2     | 1     | 0     |
|----------------|---|---|---|-------|-------|-------|-------|-------|
|                | - | - | 1 | STRC4 | STRC3 | STRC2 | STRC1 | STRC0 |
| 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   |

|        | <b>5</b> 2.11 | Initial | <b>5</b> 244                 |  |
|--------|---------------|---------|------------------------------|--|
| Bit    | Bit Name      | Value   | R/W                          | Description  |
| 7 to 5 | _             | All 0   | R                            | Reserved   |
|        |               |         |                              | These bits are always read as 0. The write value should always be 0.   |
| 4      | STRC4         | 0       | R/W                          | Counter C4 Start   |
| 3      | STRC3         | 0       | R/W                          | These bits enable and disable timer counter Cn (TCNTCn)  |
| 2      | STRC2         | 0       | R/W                          | in a subblock.   |
| 1      | STRC1         | 0       | R/W                          | <ul> <li>─ When bit STRCn is cleared to 0, TCNTCn is stopped.</li> <li>_ While TCNTCn is stopped, the previous counter value is</li> </ul>   |
| 0      | STRC0         | 0       | While TCNTCh is stopped, the | retained and TCNTCn is resumed from the value when   |
|        |               |         |                              | Note that counting is enabled when both this bit and the TCE bit in ATUENR must be set to 1.   |
|        |               |         |                              | 0: TCNTCn is disabled  |
|        |               |         |                              | 1: TCNTCn is enabled   |
|        |               |         |                              | Note: The prescalers run regardless of this bit and are not synchronized with the timing at which this bit is set. Therefore, the time from when this bit is set to 1 until TCNTCn is incremented for the first time is less than the cycle time of the clock of TCNTCn. |

Note: n = 0 to 4

### 13.16.2 Noise Canceler Control Register C0 to C4 (NCCRC0 to NCCRC4)

NCCRC0 to NCCRC4 is an 8-bit readable/writable register. The noise cancellation is performed on the input capture trigger signal input from pin TIOCnm in subblock C0 to C4 by setting this register. Two modes are available in noise cancellation and can be switched by bit NCMCn.

NCCRC0 to NCCRC4 can be read from and written to in byte or word units.

NCCRC0 to NCCRC4 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3          | 2          | 1          | 0          |
|----------------|---|---|---|---|------------|------------|------------|------------|
|                | 1 | - | - | - | NCEC<br>n3 | NCEC<br>n2 | NCEC<br>n1 | NCEC<br>n0 |
| Initial value: | 0 | 0 | 0 | 0 | 0          | 0          | 0          | 0          |
| R/W:           | R | R | R | R | R/W        | R/W        | R/W        | R/W        |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 4 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |

|     |          | Initial |     |  |
|-----|----------|---------|-----|--|
| Bit | Bit Name | Value   | R/W | Description  |
| 3   | NCECn3   | 0       | R/W | Noise Canceler Enable Cn3 to Cn0                           |
| 2   | NCECn2   | 0       | R/W | These bits enable and disable the noise cancelers for pins |
| 1   | NCECn1   | 0       | R/W | TIOCnm in subblock C0 to C4.                               |
| 0   | NCECn0   | 0       | R/W | O: Noise cancelers for inputs on TIOCnm are disabled       |
|     |          |         |     | Noise cancelers for inputs on TIOCnm are enabled           |

When a level change on externally input signals TIOCnm is detected while this bit is set to 1, it is processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the noise cancellation mode register (NCMR) of the common controller.

In premature-transition cancellation mode

When a level change of the externally input signal is detected, the change is output as the signal whose noise is removed and the corresponding noise canceler counter (NCNTCnm) is started for counting up. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRCnm). The level of the externally input signal is output on this compare match.

When these bits are cleared to 0 while the counter (NCNTCnm) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

In minimum time-at-level cancellation mode

When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTCnm) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRCnm), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed.

When these bits are cleared to 0 while the counter (NCNTCnm) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected.

For details on operations in each mode, see figures 13.1 and 13.2.

Note: n = 0 to 4, m = 0 to 3

## 13.16.3 Timer Control Registers C0 to C4 (TCRC0 to TCRC4)

TCRC0 to TCRC4 are 8-bit readable/writable registers that select the counter clock for subblocks C0 to C4 and enable and disable the PWM mode and the forced compare matches.

TCRC0 to TCRC4 can be read from and written to in byte or word units.

TCRC0 to TCRC4 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 6        |            | 5          | 4          | 3         | 2 1 |        | 0    |
|----------------|------------|------------|------------|------------|-----------|-----|--------|------|
|                | FCMC<br>n3 | FCMC<br>n2 | FCMC<br>n1 | FCMC<br>n0 | PWM<br>n0 | CKS | SELCn[ | 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  |

Note: n = 0 to 4 (correspond to subblocks C0 to C4)

| Bit | Bit Name   | Initial<br>Value  | R/W  | Description  |
|-----|--|---|--|--|
| 7   | FCMCn3   | 0   | R/W  | Forced Compare Match Cnm   |
|     |  |   |  | <u> </u>   |
| 6   | FCMCn2   | 0   | R/W  | Setting these bits to 1 generates forced compare match when a general register (GRCnm) is used for compare   |
| 5   | FCMCn1   | 0   | R/W  | _ match.   |
| 4   | FCMCn0   | 0   | R/W  | 0: No forced compare match occurs on general register GRCnm  |
|     |  |   |  | Forced compare match generated on general register<br>GRCnm  |
|     |  |   | A compare match flag (IMFCnm) in timer status register Cn (TSRCn) is set to 1 in the cycle following the cycle in which the FCMCnm bit is set to 1. An output level on the TIOCnm signal is also changed in a similar way of normal compare match. |  |
|     | Who control of the co | While this bit is set to 1, the state of a compare match occurrence is retained regardless of GRCnm and TCNTCn. TCNTCn continues counting up on its counting clock. |  |  |
|     |  |   |  | When the compare match flag bit (IMFCnm) in timer status register Cn (TSRCn) is cleared to 0, this bit is automatically cleared to 0. Any of compare match on a general register and a counter is ignored until this bit is cleared.   |
|     |  |   |  | After this bit is cleared to 0, compare match of TCNTCn and GRCnm can be performed.  |
|     |  |   |  | When both bits PWMn0 and FCMCn0 are set to 1 and forced compare match occurs. At this time, a counter is cleared and a signal is output on pin TIOCnm according to the value in the IOCn0 bits. TCNTCn restarts counting up from H'000000. After that, while bit FCMCn0 is set to 1, another compare match is not generated. (Only a counter of channel 0 in each subblock can be cleared on compare match.) |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 3   | PWMn0    | 0                | R/W | PWM Mode  |
|     |          |                  |     | Setting this bit to 1 makes subblock Cn operate in PWM mode. In PWM mode, TCNTCn is cleared on compare match between TCNTCn and general register GRCn0. The setting of this bit is valid when GRCn0 functions as a compare match register.  |
|     |          |                  |     | When general register GRCnm (m = 1 to 3) functions as a compare match register, a signal is output on pin TIOCnm according to the setting in the IOCnm bits in TIORCn.  |
|     |          |                  |     | A signal level of a logical one can be output on cycle compare match with GRCn0, and a signal level of a logical zero can be output on duty cycle compare match with GRCn1 to GRCn3.  |
|     |          |                  |     | To make the subblock operate in PWM mode, further setting is needed. Select the compare match by the IOCnm bits in timer I/O control register C (TIORCn) for GRCn1 to GRCn3 which operate in PWM mode and GRCn0. Note that a logical zero output on compare match must be selected for GRCn 1 to GRCn3. |
|     |          |                  |     | 0: Subblock Cn does not operate in PWM mode   |
|     |          |                  |     | 1: Subblock Cn operates in PWM mode   |
|     |          |                  |     | When TCNTCn matches GRCn0 while this bit is set to 1, TCNTCn is cleared. However, when clearing the counter on compare match and incrementation occur simultaneously, TCNTCn is set to H'000001. This occurs when TCNTCn is driven by the clock whose frequency is equal to the P $\phi$ clock.         |
|     |          |                  |     | In PWM mode, do not set GRCn0 to GRCn3 to H'000000. If GRCn0 is set to H'000000, compare match occurs at illegal cycles.  |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 2 to 0 | CKSELCn  | 000              | R/W | TCNTCn Clock Select  |
|        | [2:0]    |                  |     | These bits select the counter clock of subblock Cn. Counters in subblock Cn (TCNTCn) are driven by the clock selected in these bits. |
|        |          |                  |     | 000: Counters are driven by clock-bus line 0   |
|        |          |                  |     | 001: Counters are driven by clock-bus line 1   |
|        |          |                  |     | 010: Counters are driven by clock-bus line 2   |
|        |          |                  |     | 011: Counters are driven by clock-bus line 3   |
|        |          |                  |     | 100: Counters are driven by clock-bus line 4   |
|        |          |                  |     | 101: Counters are driven by clock-bus line 5   |
|        |          |                  |     | 11x: Reserved  |
|        |          |                  |     | x denotes don't care.  |

Note: n = 0 to 4 (n denotes subblock number), m = 0 to 3 (m denotes general register number)

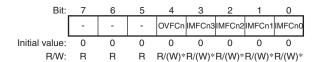
### 13.16.4 Timer Status Registers C0 to C4 (TSRC0 to TSRC4)

TSRC0 to TSRC4 are 8-bit readable/writable registers indicate occurrence of overflow of timer counter Cn (TCNTCn) in subbblocks C0 to C4 and input capture and compare match of GRC00 to GRC43.

The flags are in this register are used as interrupts. When the corresponding bit in timer interrupt enable register C (TIERC) is set to 1, an interrupt request is issued.

TSRC0 to TSRC4 can be read from and written to in byte or word units.

TSRC0 to TSRC4 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Note: n = 0 to 4 (correspond to subblocks C0 to C4)

<sup>\*</sup> Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

| Bit    | Bit<br>Name | Initial<br>Value | R/W | Description   |
|--------|-------------|------------------|-----|---|
| 7 to 5 | _           | All 0            | R   | Reserved  |
|        |             |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 4      | OVFCn       | 0                | R/W | Overflow Flag Cn  |
|        |             |                  |     | Indicates whether or not TCNTCn has overflowed. This flag cannot be set to 1 by software.   |
|        |             |                  |     | 0: TCNTCn has not overflowed  |
|        |             |                  |     | [Clearing condition]  |
|        |             |                  |     | When writing 0 to this bit after reading it as 1  |
|        |             |                  |     | 1: TCNTCn has overflowed  |
|        |             |                  |     | [Setting condition]   |
|        |             |                  |     | <ul> <li>When TCNTCn overflows (from H'FFFFFF to<br/>H'000000)</li> </ul>   |
|        |             |                  |     | When TCNTCn is incremented while it is H'FFFFF, it overflows. When writing H'000000 to TCNTCn or TCNTCn is started from H'000000, this bit is not set to 1. |
|        |             |                  |     | When writing to TCNTCn at the same time as incrementation while it is H'FFFFFF, this bit is set to 1. However, TCNTCn is started from the written value.    |

| Bit    | Bit<br>Name  | Initial<br>Value | R/W | Description   |
|--------|--------------|------------------|-----|---|
| 3 to 0 | IMFCn3       | All 0            | R/W | Input Capture/Compare Match Flag Cnm  |
|        | to<br>IMFCn0 |                  |     | These bits indicate whether or not input capture and compare match between general register GRCnm and TCNTCn has occurred. This flag cannot be set to 1 by software. Setting and clearing conditions are shown below. |
|        |              |                  |     | 0: Neither input capture nor compare match has occurred   |
|        |              |                  |     | [Clearing conditions: input capture/output compare]   |
|        |              |                  |     | <ul> <li>When writing 0 to this bit after reading it as 1</li> </ul>  |
|        |              |                  |     | <ul> <li>When bits in subblocks C00, C10, C20, C30, and C40<br/>are cleared by the ACK signal output from the DMAC<br/>[Clearing condition: input capture]</li> </ul>   |
|        |              |                  |     | When this bit is automatically cleared on read access   |
|        |              |                  |     | to general register GRCnm by the A-DMAC   |
|        |              |                  |     | 1: Input capture or compare match has occurred  |
|        |              |                  |     | [Setting condition: input capture]  |
|        |              |                  |     | <ul> <li>When GRCnm functions as an input capture register<br/>and the value in TCNTCn is transferred to GRCnm on<br/>an assertion of the input capture signal</li> </ul>   |
|        |              |                  |     | [Setting conditions: output compare]  |
|        |              |                  |     | <ul> <li>When GRCnm functions as an compare match<br/>register and the values in TCNTCnm and GRCnm<br/>match</li> </ul>   |
|        |              |                  |     | <ul> <li>When the forced compare match bit (FCMCnm) in<br/>TCRCn is set to 1</li> </ul>   |
|        |              |                  |     | Even if these bits are set to 1 meaning that the flag has<br>not been cleared, another input capture or output compare<br>signal can be input. A value of 1 is written to these bits.                                 |
|        |              |                  |     | Even if the compare match flag is cleared to 0 while TCNTCn = GRCnm after the compare match is detected, these bits are not set to 1.   |
|        |              |                  |     | Bits in subblocks C00, C10, C20, C30, and C40 can be used as the interrupt request to the D-MAC. These bits can be automatically cleared to 0 by the ACK signal.  |

Notes: n = 0 to 4 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

\* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

## 13.16.5 Timer Interrupt Enable Registers C0 to C4 (TIERC0 to TIERC4)

TIERC0 to TIERC4 are 8-bit readable/writable registers that enable and disable interrupt requests for timer C as the input capture, output compare, and overflow-interrupt.

TIERC0 to TIERC4 can be read from and written to in byte or word units.

TIERC0 to TIERC4 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4     | 3      | 2      | 1      | 0      |
|----------------|---|---|---|-------|--------|--------|--------|--------|
|                | - | - | - | OVECn | IMECn3 | IMECn2 | IMECn1 | IMECn0 |
| 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    |

Note: n = 0 to 4 (correspond to subblocks C0 to C4)

| Bit    | Bit<br>Name | Initial<br>Value | R/W | Description   |
|--------|-------------|------------------|-----|---|
| 7 to 5 | _           | All 0            | R   | Reserved  |
|        |             |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 4      | OVECn       | 0                | R/W | Overflow Interrupt Enable Cn  |
|        |             |                  |     | Enables and disables an interrupt request when the overflow flag (OVFCn) in timer status register Cn (TSRCn) is set to 1. |
|        |             |                  |     | 0: Disables an OVFCn interrupt request  |
|        |             |                  |     | 1: Enables an OVFCn interrupt request   |
| 3      | IMECn3      | 0                | R/W | Input Capture/Compare Match Interrupt Enable nm   |
| 2      | IMECn2      | 0                | R/W | These bits enable and disable an interrupt request when   |
| 1      | IMECn1      | 0                | R/W | <sup>−</sup> the input capture/compare match flag in timer status<br>_ register Cn (TSRCn).                               |
| 0      | IMECn0      | 0                | R/W | 0: Disables an IMFCnm interrupt request   |
|        |             |                  |     | 1: Enables an IMFCnm interrupt request  |

Note: n = 0 to 4 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

# 13.16.6 Timer I/O Control Registers C0 to C4 (TIORC0 to TIORC4)

TIORC0 to TIORC4 are 16-bit readable/writable registers.

Compare match and input capture functions are switched by the setting in the IOCnm[2] bit. Before changing the IOCnm[2] bit, clear the IOCnm[1:0] bits to B'00. After that, the output signal level for compare match or the input capture edge for input capture can be set.

When the compare match function is selected, the initial level of the compare match signal is a logical zero. During operation, the previous level is retained. When two functions are switched, the counter must be stopped. Otherwise, operation cannot be guaranteed.

TIORC0 to TIORC4 can be read from and written to in byte- or word-units.

TIORC0 to TIORC4 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 15 | 14  | 13      | 12  | 11 | 10  | 9       | 8   | 7 | 6   | 5      | 4   | 3 | 2   | 1      | 0   |
|----------------|----|-----|---------|-----|----|-----|---------|-----|---|-----|--------|-----|---|-----|--------|-----|
|                | 1  | IC  | )Cn3[2: | 0]  | 1  | IC  | )Cn2[2: | :0] | 1 | IC  | )Cn1[2 | :0] | - | IC  | )Cn0[2 | :0] |
| Initial value: | 0  | 0   | 0       | 0   | 0  | 0   | 0       | 0   | 0 | 0   | 0      | 0   | 0 | 0   | 0      | 0   |
| R/W:           | R  | R/W | R/W     | R/W | R  | R/W | R/W     | R/W | R | R/W | R/W    | R/W | R | R/W | R/W    | R/W |

Note: n = 0 to 4 (correspond to subblocks C0 to C4)

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 15  | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0. |

| Bit      | Bit Name   | Initial<br>Value | R/W | Description   |
|----------|------------|------------------|-----|---|
| 14 to 12 | IOCn3[2:0] | 000              | R/W | I/O Control   |
|          |            |                  |     | These bits select the function of general register GRCn3.         |
|          |            |                  |     | When GRCn3 functions as the output compare register               |
|          |            |                  |     | 000: Compare match function disabled                              |
|          |            |                  |     | 001: Logical zero is output on compare match                      |
|          |            |                  |     | 010: Logical one is output on compare match                       |
|          |            |                  |     | 011: Output levels are toggled every compare match                |
|          |            |                  |     | When GRCn3 functions as the input capture register                |
|          |            |                  |     | 100: Input capture function disabled                              |
|          |            |                  |     | 101: Input capture on the rising edge of TIOCnm                   |
|          |            |                  |     | 110: Input capture on the falling edge of TIOCnm                  |
|          |            |                  |     | 111: Input capture on both edges of TIOCnm                        |
| 11       | _          | 0                | R   | Reserved  |
|          |            |                  |     | This bit is always read as 0. The write value should always be 0. |
| 10 to 8  | IOCn2[2:0] | 000              | R/W | I/O Control   |
|          |            |                  |     | These bits select the function of general register GRCn2.         |
|          |            |                  |     | When GRCn2 functions as the output compare register               |
|          |            |                  |     | 000: Compare match function disabled                              |
|          |            |                  |     | 001: Logical zero is output on compare match                      |
|          |            |                  |     | 010: Logical one is output on compare match                       |
|          |            |                  |     | 011: Output levels are toggled every compare match                |
|          |            |                  |     | When GRCn2 functions as the input capture register                |
|          |            |                  |     | 100: Input capture function disabled                              |
|          |            |                  |     | 101: Input capture on the rising edge of TIOCnm                   |
|          |            |                  |     | 110: Input capture on the falling edge of TIOCnm                  |
|          |            |                  |     | 111: Input capture on both edges of TIOCnm                        |
| 7        | _          | 0                | R   | Reserved  |
|          |            |                  |     | This bit is always read as 0. The write value should always be 0. |

|        |            | Initial |     |   |
|--------|------------|---------|-----|---|
| Bit    | Bit Name   | Value   | R/W | Description   |
| 6 to 4 | IOCn1[2:0] | 000     | R/W | I/O Control   |
|        |            |         |     | These bits select the function of general register GRCn1.         |
|        |            |         |     | When GRCn1 functions as the output compare register               |
|        |            |         |     | 000: Compare match function disabled                              |
|        |            |         |     | 001: Logical zero is output on compare match                      |
|        |            |         |     | 010: Logical one is output on compare match                       |
|        |            |         |     | 011: Output levels are toggled every compare match                |
|        |            |         |     | When GRCn1 functions as the input capture register                |
|        |            |         |     | 100: Input capture function disabled                              |
|        |            |         |     | 101: Input capture on the rising edge of TIOCnm                   |
|        |            |         |     | 110: Input capture on the falling edge of TIOCnm                  |
|        |            |         |     | 111: Input capture on both edges of TIOCnm                        |
| 3      | _          | 0       | R   | Reserved  |
|        |            |         |     | This bit is always read as 0. The write value should always be 0. |
| 2 to 0 | IOCn0[2:0] | 000     | R/W | I/O Control   |
|        |            |         |     | These bits select the function of general register GRCn0.         |
|        |            |         |     | When GRCn0 functions as the output compare register               |
|        |            |         |     | 000: Compare match function disabled                              |
|        |            |         |     | 001: Logical zero is output on compare match                      |
|        |            |         |     | 010: Logical one is output on compare match                       |
|        |            |         |     | 011: Output levels are toggled every compare match                |
|        |            |         |     | When GRCn0 functions as the input capture register                |
|        |            |         |     | 100: Input capture function disabled                              |
|        |            |         |     | 101: Input capture on the rising edge of TIOCnm                   |
|        |            |         |     | 110: Input capture on the falling edge of TIOCnm                  |
|        |            |         |     | 111: Input capture on both edges of TIOCnm                        |

Note: n = 0 to 4 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

# 13.16.7 Timer Counters C0 to C4 (TCNTC0 to TCNTC4)

TCNTC0 to TCNTC4 are 32-bit readable/writable registers driven by the input clock. These counters can be read from and written to while they are being run.

Timer counter Cn (TCNTCn) is started for counting by setting the bit in timer start register C (TSTRC) to 1. The clock signal is selected by the clock select bit (CKSEL) in timer control register Cn (TCRCn). When these counters overflow, the overflow flag (OVFCn) in timer status register Cn (TSRCn) is set to 1.

TCNTC0 to TCNTC4 can be read from and written to in longword units.

TCNTC0 to TCNTC4 are initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

### 13.16.8 General Registers C00 to C43 (GRC00 to GRC43)

GRC00 to GRC43 are 32-bit readable/writable registers that function as the input capture register or output compare register. These functions are switched by setting timer I/O control register Cn (TIORCn).

If these registers function as the input capture register, they capture the value in TCNTCn when the externally input capture signal is detected. At this time, the IMFC bit in timer status register Cn (TSRCn) is set to 1. The edge to be detected is selected by TIORCn.

Input capture is performed even if the counter is stopped (the TCE bit in ATUENR is cleared to 0 or the STRCn bit in TSTRC is cleared to 0). The value in the counter stopped is loaded to GRCnm.

If these registers function as the output compare register, the values in GRCnm and the timer counter (TCNTCn) are constantly compared. The IMFC bit in timer status register (TSRCn) is set to 1 on the first edge of the  $P\phi$  clock after they match. At this time, the level on the TIOCnm pin is changed according to the setting in TIORCn (logical zero, logical one, or toggled). Initially, logical zero is output on the TIOCnm pin (immediately after a reset or output compare modes are switched). During operation, the previous value is output.

GRC00 to GRC43 can be read from and written to in longword units.

GRC00 to GRC43 are initialized to H'FFFF FF00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

### 13.16.9 Noise Canceler Counters C00 to C43 (NCNTC00 to NCNTC43)

NCNTC00 to NCNTC43 are 8-bit readable/writable registers.

NCNTC00 to NCNTC43 are started by the external input signals TIOC00 to TIOC43 as triggers when the noise canceler is enabled by the noise canceler enable bit (NCEC00 to NCEC43) in timer I/O control register C0 to C4 (TIORC0 to TIORC4). These counters are driven by the  $P\phi$  clock or the  $P\phi$  clock divided by 128 output from the prescaler.

NCNTCnm continues counting regardless of the settings in the TCE bit in ATU-III master enable register (ATUENR) and in TSTRC (regardless of the TCNTCn).

The input signals are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting of the timer C noise cancellation mode bit in the noise cancellation mode register (NCMR) of the common controller.

#### Premature-Transition Cancellation Mode

When a level change of the externally input signal (TIOCnm) is detected while bit NCECnm is set to 1 and NCNTCnm is stopped, NCNTCnm is started for counting up. This counter is cleared to H'00 and stopped on the first edge of the P\$\phi\$ clock after the value in NCNTCnm matches the value in noise canceler register nm (NCRCnm). NCNTCnm is incremented regardless of the TCE bit in ATUENR. The first change is output as the signal whose noise is removed and the edge is to be extracted. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRCnm). The level of the externally input signal is output on this compare match.

When the NCECnm bit is cleared to 0 while the counter (NCRCnm) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

• In Minimum Time-at-Level Cancellation Mode

When a level change of the externally input signal (TIOCnm) is detected while the NCECnm bit is set to 1 and NCNTnm is stopped, NCNTnm is started for counting up. This counter is cleared to H'00 and stopped on the first edge of the P $\phi$  clock after the value in NCNTnm matches the value in noise canceler registers nm (NCRCnm) or after the level of the externally input signal (TIOCnm) is changed. NCNTnm is incremented regardless of the TCE bits in ATUENR.

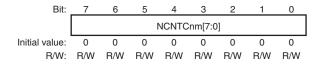
When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTnm) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRCnm), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes is treated as noise. Therefore the signal whose noise is removed is not changed.

When the NCECnm bits are cleared to 0 while the counter (NCNTCnm) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected.

NCNTCnm can be written to regardless of the operating state and can start from the value rewritten. A value greater that the value in NCRCnm is settable. In this case, NCNTCnm overflows from H'FF to H'00 and then it is cleared. After that, it is compared with NCRCnm.

NCNTC00 to NCNTC43 can be read from and written to in byte or word units.

NCNTC00 to NCNTC43 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Notes: 1. n = 0 to 4

2. m = 0 to 3 (correspond to GRC00 to GRC43)

| Bit    | Bit Name     | Initial<br>Value | R/W | Description                            |
|--------|--------------|------------------|-----|--|
| 7 to 0 | NCNTCnm[7:0] | All 0            | R/W | Noise Cancel Count Cnm                 |
|        |              |                  |     | These bits store an 8-bit count value. |

Note: n = 0 to 4, m = 0 to 3; GBC00 to GBC43

## 13.16.10 Noise Cancel Registers C00 to C43 (NCRC00 to NCRC43)

NCRC00 to NCRC43 are 8-bit readable/writable registers that are provided in each subblock and set the upper limitations of noise canceler counters C00 to C43 (NCNTC00 to NCNTC43).

A pulse width of up to 1.64 ms (=  $50 \text{ ns} \times 128 \text{ division} \times 256 \text{ counts}$ ) can be treated as noise by setting the registers to H'FF.

Input edges are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting of the NCMC bit in the noise cancellation mode register (NCMR) of the common controller.

- Premature-Transition Cancellation Mode
  - While NCNTCnm is in count operation, the level change of the subsequent input signal is masked. Values in NCNTCnm and NCRCnm are always compared. If a compare match occurs, the value in NCNTCnm is cleared on the next  $P\phi$  clock, the count operation is stopped, and the masking of the input signal is canceled.
- In Minimum Time-at-Level Cancellation Mode
   While NCNTCnm is in count operation, noise canceler processing waiting state is entered.
   Values in NCNTCnm and NCRCnm are always compared. If a compare match occurs, the
   value in NCNTCnm is cleared on the next Pφ clock, the count operation is stopped, and at the
   same time the noise canceler outputs the input signal that has passed through the noise
   canceling processing.

NCRC00 to NCRC43 can be read from and written to in byte or word units.

NCRC00 to NCRC43 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7   | 6   | 5   | 4    | 3       | 2   | 1   | 0   |
|----------------|-----|-----|-----|------|---------|-----|-----|-----|
|                |     |     |     | NCRC | nm[7:0] |     |     |     |
| Initial value: | 0   | 0   | 0   | 0    | 0       | 0   | 0   | 0   |
| R/W:           | R/W | R/W | R/W | R/W  | R/W     | R/W | R/W | R/W |
|                |     |     |     |      |         |     |     |     |

Notes: 1. n = 0 to 4 2. m = 0 to 3 (correspond to GRC00 to GRC43)

| Bit    | Bit Name    | Initial<br>Value | R/W | Description                                      |
|--------|-------------|------------------|-----|--|
| 7 to 0 | NCRCnm[7:0] | All 0            | R/W | Noise Cancellation Time Cnm                      |
|        |             |                  |     | TIOCnm noise cancel period (8-bit compare value) |

#### **Operations of Timer C** 13.17

#### 13.17.1 **Input Capture Function**

General registers GRCnm of timer C perform input capture operation when the input capture operation is selected by timer I/O control register Cn (TIORCn). Capturing is perfumed when an edge on external input pins (TIOCnm) is detected.

Timer counter Cn (TCNTCn) is started for counting up by setting a bit in timer start register C (TSTRC) to 1. When an edge on an external pin corresponding to GRCnm is detected, a bit (IMFCnm) in timer status register C (TSRC) is set to 1 and the counter value is transferred to GRCnm. This flag indicating that an input capture has occurred and an interrupt request signal are changed two cycles of the P\psi clock after the edge on pin TIOCnm is detected.

The edge type is selected from the rising, falling, or both by the IOCnm[2:0] bits in TIORCn. An interrupt request can be output by setting timer interrupt enable register Cn (TIERCn).

The input capture flag (the IMFCnm bits in TSRCn) is cleared; when 0 is written to the bits after they are read as 1; when GRCnm is read by the A-DMAC (automatic clearing); when an ACK signal is input by the DMAC.

Figure 13.24 shows an operation example of input capture in subblock C0. In this example, the both edges on pin TIOC00, the rising edge on pin TIOC01, and the falling edge on TIOC02 are selected.

Input capture is performed even if TCNTCn is stopped (when the TCE bit in ATUENR or bit STRCn is cleared to 0). The counter value of the TCNTCn stopped is captured in GRCnm.

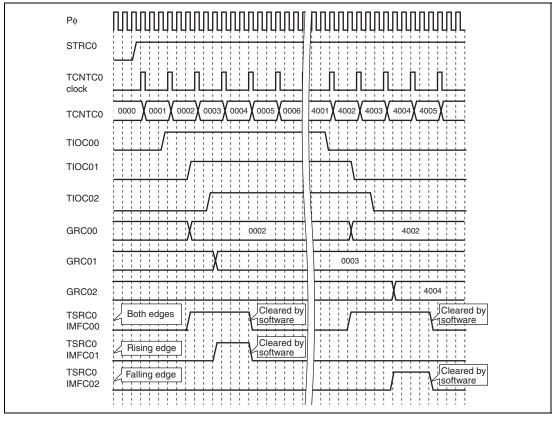


Figure 13.24 Operation Example of Input Capture in Subblock C0

# 13.17.2 Compare Match Function

A compare match signal can be output on external pin TIOCnm by setting general register GRCnm as the compare match register in timer I/O control register Cn (TIORCn).

Timer counter Cn (TCNTCn) is started for counting up by setting a bit in timer start register C (TSTRC) to 1. Set the value in GRCnm before starting the counter. When the values in GRCnm and TCNTCn match, a bit corresponding to GRCnm in timer status register Cn (TSRCn) is set and a waveform is output on external pin TIOCnm.

The compare match flag is set and the signal level on pin TIOCnm is changed on the first edge of the P $\phi$  clock immediately after compare match between GRCnm and TCNTCn.

A logical one, a logical zero, or a toggled output can be selected for the signal to be output on pin TIOCnm.

An interrupt can be output by setting timer interrupt enable register C (TIERC).

Figure 13.25 shows an operation example of compare match. In this example, a toggled output on GRC00, a logical one output on GRC01, and a logical zero output on GRC02 are externally output. H'004004 is set in GRC0m. The status flag and the output level on pin TIOC0m are changed on the first edge of the P\u03c9 clock immediately after compare match between GRCnm and TCNTCn.

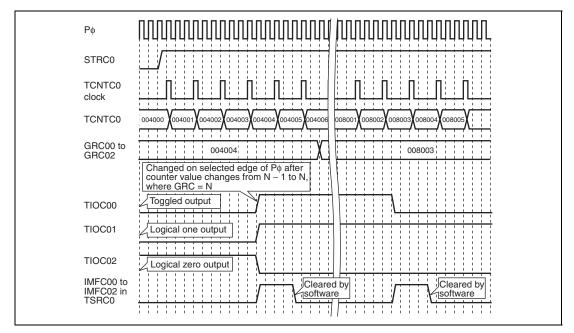


Figure 13.25 Operation Example of Compare Match

By setting a forced compare match bit (FMCMnm) in timer control register Cn (TCRCn), compare match can be generated even if TCNTCn has not matched GRCnm. The compare match flag and the signal level on pin TIOCnm are changed on the first edge of the P $\phi$  clock after bit FCMCnm is set to 1. Compare match of TCNTCn with GRCnm after a forced compare match is not performed until the compare match flag (IMFCnm) is cleared. Bit FCMCnm is cleared at the same time as bit IMFCnm in TSRCn is cleared.

To clear the compare match flag, write 0 to the status flag after it is read as 1. GRC00, GRC10, GRC20, GRC30, and GRC40 can be cleared by an interrupt request from the DMAC.

Compare match is detected when any of the following occurs.

When the values in TCNTCn and GRCnm match (other than after forced compare match)

- When the forced compare match bit (bit FCMCnm in TCRCn) is changed from 0 to 1
- When the values in TCNTCn and GRCnm match because a counter is cleared on compare match of GRCn0

The compare match flag and the signal level on pin TIOCnm is changed on the first edge of the  $P\phi$  clock after bit FCMCnm is set to 1. Make sure that the compare match operation is selected by bits IOCnm[2:0] in TIORCn before starting operation. Compare match is not detected; when GRCnm is set to the same value as TCNTCn; when the compare match operation is selected after setting the forced compare match bit to 1.

Compare match is detected regardless of the counter operating state. Even if the counter is stopped, compare match occurs when the condition is satisfied.

When the compare match status flag is cleared before GRCnm and TCNTCn are changed (such as before counting while the counter has been stopped), the compare match is not detected.

#### **13.17.3 PWM Function**

Setting bit PWMn in timer control register Cn (TCRCn) to 1 makes channels 1 to 3 in each subblock function as PWM timers with the same frequency. In PWM mode, GRCn0 as a cycle setting register and GRCn1 to GRCn3 as duty cycle setting registers are used. External pins TIOCn1 to TIOCn3 corresponding GRCn1 to GRCn3 are used to output PWM signals. To use them as PWM signal outputs, select the compare match operation by bits IOCnm in TIORCn as well as setting bit PWMn so that GRCn0 to GRCn3 function as compare match registers.

Timer counter Cn (TCNTCn) is started by setting timer start register C (TSTRC). When the value in TCNTCn reaches the value in the cycle setting register (GRCn0), compare match occurs and the bit in timer status register C (TSRCn) is set to 1. At this time, TCNTCn is cleared and a signal is output on external pins TIOCn1 to TIOCn3 according to bit IOCn0 in PWM mode. The output signal levels on pin TIOCn0 depend on bit IOCn0.

When the value in TCNTCn reaches the value in the duty cycle setting register (GRCn1 to GRCn3), the bit in timer status register Cn (TSRCn) is set to 1 and a signal is output on external pins TIOCn1 to TIOCn3 according to bits IOCn1 to IOCn3.

When the same value is set in the cycle and duty cycle setting registers, priority is given to bit IOCn0 corresponding to the cycle setting register.

Figure 13.26 shows an operation example of subblock C0 in PWM mode.

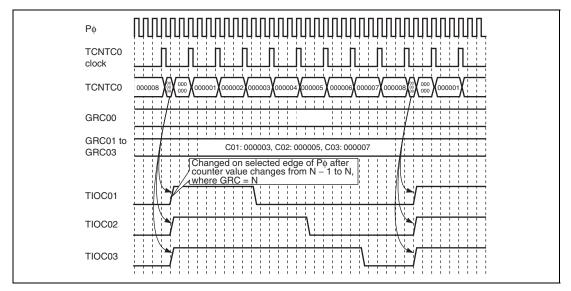


Figure 13.26 Operation Example of Subblock C0 in PWM Mode

## 13.18 Overview of Timer D

Timer D consists of four subblocks that output one-shot pulses. The subblocks are identical each other.

Timer D has the following functions.

- Down counter is started on compare match A of the output compare register, compare match B
  of the general register, or writing to the counter start bit in the down-counter start register. A
  one-shot pulse with an offset can be output.
- Output of a waveform can be forcibly terminated regardless of the down-counter value (priority is given to output termination over compare match A or writing to the counting start bit)
- General register can capture the value in TCNT2Dn on compare match A as a trigger
- Pulse indicating that compare match A or B has been detected for A/D converter activation can be output (16 lines supported by subblocks D0 and D1)
- Interrupt requests can be output on compare matches A and B (16 lines supported). The output signal is ORed on compare matches A and B.
- Interrupt requests can be output on counter overflow. Four outputs from TCNT1Dn and four outputs from TCNT2Dn.
- Interrupt requests can be output on down-counter underflow (16 lines supported). A DMA transfer request can be issued for the DMAC. In this case, the compare match flag can be cleared by the ACK signal from the DMAC. (16 lines for DMA transfer requests supported by down counters in subblocks D0 to D3)
- Offset base register can capture the counter value by a trigger from timer A

# 13.18.1 Block Diagram of Timer D

Each subblock consists of two timer counters (TCNT1Dn and TCNT2Dn), one offset base register (OSBRDn), four output compare registers (OCRDnm), four general registers (GRDnm), four timer down counters (DCNTDnm), and controller. Each channel includes two output pins; TODnmA for compare match and TODnmB for one-shot pulse output.

TODnmA and TODnmB are output a level of 0 as a default.

A trigger for activating the A/D converter can be output to the on compare matches A and B.

Figure 13.27 is a block diagram of timer D.

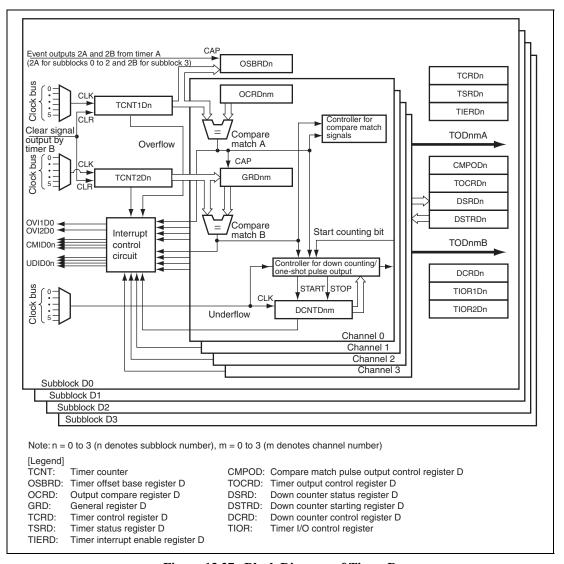


Figure 13.27 Block Diagram of Timer D

# 13.19 Description of Timer D Registers

# 13.19.1 Timer Start Register (TSTRD)

TSTRD is an 8-bit readable/writable register that enables and disables two timer counters (TCNT1Dn and TCNT2Dn) and timer down counters (DCNTDnm) in subblocks D0 to D3. When the counter Dn start bit and the TDE bit in the ATU-III master enable register (ATUENR) are both set to 1, the counters are started.

TSTRD can be read from and written to in byte or word units.

TSTRD is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3     | 2     | 1     | 0     |
|----------------|---|---|---|---|-------|-------|-------|-------|
|                | - | - | - | - | STRD3 | STRD2 | STRD1 | STRD0 |
| Initial value: | 0 | 0 | 0 | 0 | 0     | 0     | 0     | 0     |
| R/W·           | R | R | R | R | R/W   | R/W   | R/W   | R/W   |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 4 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 3      | STRD3    | 0                | R/W | Counter Dn Start   |
| 2      | STRD2    | 0                | R/W | These bits enable and disable timer counters 1Dn and 2Dn   |
| 1      | STRD1    | 0                | R/W | T(TCNT1Dn, TCNT2Dn) and timer down counters (DCNTDnm). The counter value is retained while the   |
| 0      | STRD0    | 0                | R/W | counter is stopped. When this bit is set to 1 again, the counter is restarted from the value. Note that this bit and the TDE bit in ATUENR are both set to 1 to restart the counter.   |
|        |          |                  |     | 0: TCNT1Dn, TCNT2Dn, and DCNTDnm are disabled  |
|        |          |                  |     | 1: TCNT1Dn, TCNT2Dn, and DCNTDnm are enabled   |
|        |          |                  |     | The prescalers run regardless of this counter Dn start bit and are not synchronized with the timing at which this bit is set. Therefore, the time from when this bit is set to when TCNT1Dn and TCNT2Dn are incremented for the first time is less than the cycle of the clock of TCNT1Dn and TCNT2Dn. |

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

### 13.19.2 Timer Control Registers D0 to D3 (TCRD0 to TCRD3)

TCRD0 to TCRD3 are 16-bit readable/writable registers that select the counter clocks in subblock Dn for timer counter 1 (TCNT1Dn), timer counter 2 (TCNT2Dn), and timer down counter (DCNTDnm) from clock-bus lines 0 to 5. These registers also enable and disable capture of the timer offset base register and counter clearing requests output from timer B for TCNT1Dn and TCNT2Dn.

TCRD0 to TCRD3 can be read from and written to in byte- or word-units.

TCRD0 to TCRD3 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 15 | 14         | 13         | 12         | 11 | 10  | 9     | 8     | 7 | 6   | 5     | 4     | 3 | 2   | 1     | 0     |
|----------------|----|------------|------------|------------|----|-----|-------|-------|---|-----|-------|-------|---|-----|-------|-------|
|                | 1  | OBR<br>EDn | C2C<br>EDn | C1C<br>EDn | -  | CKS | EL2Dn | [2:0] | - | CKS | EL1Dn | [2:0] | - | DC  | SELDn | [2:0] |
| Initial value: | 0  | 0          | 0          | 0          | 0  | 0   | 0     | 0     | 0 | 0   | 0     | 0     | 0 | 0   | 0     | 0     |
| R/W:           | R  | R/W        | R/W        | R/W        | R  | R/W | R/W   | R/W   | R | R/W | R/W   | R/W   | R | R/W | R/W   | R/W   |

| D:4 | Dit Name | Initial | DAM | Description   |
|-----|----------|---------|-----|---|
| Bit | Bit Name | Value   | R/W | Description   |
| 15  | _        | 0       | R   | Reserved  |
|     |          |         |     | This bit is always read as 0. The write value should always be 0.   |
| 14  | OBREDn   | 0       | R/W | Timer Offset Base Register Enable   |
|     |          |         |     | Enables and disables capture of timer offset base register Dn (OSBRDn). When this bit is set to 1, the value in TCNT1Dn is captured by OSBRDn in the P $\phi$ clock cycle following cycle in which an event from timer A is asserted. When the pulse width of the event exceeds one cycle of the P $\phi$ clock, the counter value is captured every clock cycle. |
|     |          |         |     | 0: Input capture by OSBRDn is enabled   |
|     |          |         |     | 1: Input capture by OSBRDn is disabled  |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 13  | C2CEDn   | 0                | R/W | Counter 2 Clear Enable  |
|     |          |                  |     | Enables and disables clearing the value in TCNT2Dn by timer B.  |
|     |          |                  |     | When an edge of the counter clearing signal output from timer B is detected while this bit is set to 1, TCNT2Dn is cleared in the following timing.                               |
|     |          |                  |     | When the rising edge of the clearing signal is detected in<br>the cycle in which TCNT2Dn is counted up, the counter is<br>cleared on the counting up timing.                      |
|     |          |                  |     | When the rising edge of the clearing signal is detected in other than the TCNT2Dn counting-up cycle, the counter is cleared on the first counting up timing after edge detection. |
|     |          |                  |     | The counter clearing signal is ignored with this bit set to the initial value.  |
|     |          |                  |     | 0: TCNT2Dn clearing signal from timer B is disabled   |
|     |          |                  |     | 1: TCNT2Dn clearing signal from timer B is enabled  |
| 12  | C1CEDn   | 0                | R/W | Counter 1 Clear Enable  |
|     |          |                  |     | Enables and disables clearing the value in TCNT1Dn by timer B.  |
|     |          |                  |     | When an edge of the counter clearing signal output from timer B is detected while this bit is set to 1, TCNT1Dn is cleared in the following timing.                               |
|     |          |                  |     | When the rising edge of the clearing signal is detected in<br>the cycle in which TCNT1Dn is counted up, the counter is<br>cleared on the counting up timing.                      |
|     |          |                  |     | When the rising edge of the clearing signal is detected in other than the TCNT1Dn counting-up cycle, the counter is cleared on the first counting up timing after edge detection. |
|     |          |                  |     | The counter clearing signal is ignored with this bit set to the initial value.  |
|     |          |                  |     | 0: TCNT1Dn clearing signal from timer B is disabled   |
|     |          |                  |     | 1: TCNT1Dn clearing signal from timer B is enabled  |
| 11  | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.   |

| Bit     | Bit Name | Initial<br>Value | R/W | Description   |
|---------|----------|------------------|-----|---|
| 10 to 8 | CKSEL2Dn | 000              | R/W | TCNT2Dn Clock Select  |
|         | [2:0]    |                  |     | These bits select the TCNT2Dn counting-up clock.                  |
|         |          |                  |     | 000: Incrementation of TCNT2Dn is driven by clock-bus line 0.     |
|         |          |                  |     | 001: Incrementation of TCNT2Dn is driven by clock-bus line 1.     |
|         |          |                  |     | 010: Incrementation of TCNT2Dn is driven by clock-bus line 2.     |
|         |          |                  |     | 011: Incrementation of TCNT2Dn is driven by clock-bus line 3.     |
|         |          |                  |     | 100: Incrementation of TCNT2Dn is driven by clock-bus line 4.     |
|         |          |                  |     | 101: Incrementation of TCNT2Dn is driven by clock-bus line 5.     |
|         |          |                  |     | 11x: Reserved   |
| 7       | _        | 0                | R   | Reserved  |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 6 to 4  | CKSEL1Dn | 000              | R/W | TCNT1Dn Clock Select  |
|         | [2:0]    |                  |     | These bits select the TCNT1Dn counting-up clock.                  |
|         |          |                  |     | 000: Incrementation of TCNT1Dn is driven by clock-bus line 0.     |
|         |          |                  |     | 001: Incrementation of TCNT1Dn is driven by clock-bus line 1.     |
|         |          |                  |     | 010: Incrementation of TCNT1Dn is driven by clock-bus line 2.     |
|         |          |                  |     | 011: Incrementation of TCNT1Dn is driven by clock-bus line 3.     |
|         |          |                  |     | 100: Incrementation of TCNT1Dn is driven by clock-bus line 4.     |
|         |          |                  |     | 101: Incrementation of TCNT1Dn is driven by clock-bus line 5.     |
|         |          |                  |     | 11x: Reserved   |

RENESAS

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 3      | _        | 0                | R   | Reserved  |
|        |          |                  |     | This bit is always read as 0. The write value should always be 0.   |
| 2 to 0 | DCSELDn  | 000              | R/W | DCNTDnm Clock Select  |
|        | [2:0]    |                  |     | These bits select the DCNTDnm counting-down clock. Output of one-shot pulse (TODnmB) is synchronized with the clock selected in these bits. |
|        |          |                  |     | 000: Decrementation of DCNTDnm is driven by clock-bus line 0.   |
|        |          |                  |     | 001: Decrementation of DCNTDnm is driven by clock-bus line 1.   |
|        |          |                  |     | 010: Decrementation of DCNTDnm is driven by clock-bus line 2.   |
|        |          |                  |     | 011: Decrementation of DCNTDnm is driven by clock-bus line 3.   |
|        |          |                  |     | 100: Decrementation of DCNTDnm is driven by clock-bus line 4.   |
|        |          |                  |     | <ol><li>101: Decrementation of DCNTDnm is driven by clock-bus line 5.</li></ol>   |
|        |          |                  |     | 11x: Reserved   |

# [Legend]

x: Don't care

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number). Counters in channels 0 to 3 of the same subblock use the same clock signal.

# 13.19.3 Timer I/O Control Registers 1D0 to 1D3 (TIOR1D0 to TIOR1D3)

TIOR1D0 to TIOR1D3 are 16-bit readable/writable registers that select the source of compare match output (TODnmA), enable and disable compare match of OCRDnm, and set the output level on pin TODnA on compare match A.

TIOR1D0 to TIOR1D3 can be read from and written to in byte- or word-units.

TIOR1D0 to TIOR1D3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 15   | 14      | 13   | 12      | 11   | 10      | 9    | 8       | 7     | 6       | 5     | 4       | 3     | 2       | 1     | 0       |
|----------------|------|---------|------|---------|------|---------|------|---------|-------|---------|-------|---------|-------|---------|-------|---------|
|                | OSSD | n3[1:0] | OSSD | n2[1:0] | OSSD | n1[1:0] | OSSD | n0[1:0] | IOADr | n3[1:0] | IOADr | n2[1:0] | IOADr | n1[1:0] | IOADr | n0[1:0] |
| Initial value: | 0    | 0       | 0    | 0       | 0    | 0       | 0    | 0       | 0     | 0       | 0     | 0       | 0     | 0       | 0     | 0       |
| R/W:           | R/W  | R/W     | R/W  | R/W     | R/W  | R/W     | R/W  | R/W     | R/W   | R/W     | R/W   | R/W     | R/W   | R/W     | R/W   | R/W     |

| Bit    | Bit Name    | Initial<br>Value | R/W | Description   |
|--------|-------------|------------------|-----|---|
| 15, 14 | OSSDn3[1:0] | 00               | R/W | Compare Match Output Source Select  |
| 13, 12 | OSSDn2[1:0] | 00               | R/W | These bits select the output level on the compare match   |
| 11, 10 | OSSDn1[1:0] | 00               | R/W | output pin (TODnmA). The output is controlled by compare match A, compare match B, or both matches.   |
| 9, 8   | OSSDn0[1:0] | 00               | R/W | Example 1   |
|        |             |                  |     | Compare match A or B is used as a trigger of timer down counter D (DCNTDnm). The other compare match is used as a trigger of compare match output.  |
|        |             |                  |     | Example 2   |
|        |             |                  |     | Compare matches A or B is used as a trigger of an assertion of the output. The other compare match is used as a trigger of an negation of the output. A one-shot pulse can be output on pin TODnmA. |
|        |             |                  |     | When both matches are used as a trigger of the output and they occur at the same time, priority is given to compare match B and the output level depends on the IOBnm bit in TIOR2Dn.               |
|        |             |                  |     | A level of 0 is output on TODnmA as a default. While these bits are set to B'00, the output level on TODnmA is not changed even if compare match A or B occurs.                                     |
|        |             |                  |     | 00: Output level on TODnmA is not changed   |
|        |             |                  |     | 01: Output level on TODnmA depends on the I/O control bit A on compare match A  |
|        |             |                  |     | <ol> <li>Output level on TODnmA depends on the I/O control<br/>bit B on compare match B</li> </ol>  |
|        |             |                  |     | 11: Output level on TODnmA depends on the I/O control bit A or B on compare match A or B, respectively  |

| Bit  | Bit Name    | Initial<br>Value | R/W | Description   |
|------|-------------|------------------|-----|---|
| 7, 6 | IOADn3[1:0] | 00               | R/W | I/O Control A   |
| 5, 4 | IOADn2[1:0] | 00               | R/W | These bits select the function of the output compare  |
| 3, 2 | IOADn1[1:0] | 00               | R/W | register (OCRDnm).  |
| 1, 0 | IOADn0[1:0] | 00               | R/W | When these bits are set to B'00, compare match between OCRDnm and timer counter 1 (TCNT1Dn) is not performed. Otherwise, the compare match is performed. When the CMEADnm bit in timer interrupt enable register (TIERDn) is set to 1, an interrupt request is issued on compare match. When the CMPADnm bit in the compare match pulse output control register is set to 1, a trigger to activate the A/D converter is output (supported only by subblocks D0 and D1). |
|      |             |                  |     | If compare match A is selected by the compare match output source select bit (OSSDnm), a signal is output on pin TODnmA according to the IOADnm bits.   |
|      |             |                  |     | 00: Compare match is not performed  |
|      |             |                  |     | 01: Output level on compare match is 0  |
|      |             |                  |     | 10: Output level on compare match is 1  |
|      |             |                  |     | 11: Output level on compare match is toggled  |

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

## 13.19.4 Timer I/O Control Registers 2D0 to 2D3 (TIOR2D0 to TIOR2D3)

TIOR2D0 to TIOR2D3 are 16-bit readable/writable registers that select the function of general registers (GRDnm). GRDnm can function as capture or compare match registers. TIOR2Dn also enables and disables compare match and set the output level on pin TODnA on compare match.

The GRDnm function can be selected from the capture or compare match by bit I/O control B (IOBDnm).

When TIOR2Dn is used as the compare match register (IOBDnm[2] = 0) and the IOBDnm[1:0] bits are set to B'00, compare match between GRDnm and timer counter 2 (TCNT2Dn) is not performed. Otherwise, the compare match is performed. When the CMEBDnm bit in timer interrupt enable register (TIERDn) is set to 1, an interrupt request is issued on compare match. When the CMPBDnm bit in the compare match pulse output control register is set to 1, a trigger to activate the A/D converter is output (supported only by subblocks D0 and D1).

When compare match B is selected as the output source in the compare match output source select bit (OSSDnm), a signal is output on pin TODnmA according to the IOBDnm bits.

When TIOR2Dn is used as the capture register (IOBDnm[2:0] = B'101), it captures the value in timer counter 2 (TCNT2Dn) on compare match A. Even if TCNT2Dn is stopped, TIOR2D captures the value in TCNT2Dn on compare match A. However, when IOBDnm[2:0] = B'100, TIOR2D does not capture.

When the function is changed while TCNT2Dn = GRDnm, compare match B does not occur. For example, when a general register captures the value in TCNT2Dn, compare match B does not occur even if the function is changed.

When a general register is used as the compare match register and compare match B is selected by the OSSDnm bit, output level on pin TODnA is not changed even if the function is changed to capture, and vice versa.

TIOR2D0 to TIOR2D3 can be read from and written to in byte- or word-units.

TIOR2D0 to TIOR2D3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 15 | 14  | 13     | 12   | 11 | 10  | 9      | 8    | 7 | 6   | 5      | 4   | 3 | 2   | 1      | 0    |
|----------------|----|-----|--------|------|----|-----|--------|------|---|-----|--------|-----|---|-----|--------|------|
|                | -  | Ю   | BDn3[2 | ::0] | 1  | 10  | BDn2[2 | 2:0] | - | Ю   | BDn1[2 | :0] | - | Ю   | BDn0[2 | 2:0] |
| Initial value: | 0  | 0   | 0      | 0    | 0  | 0   | 0      | 0    | 0 | 0   | 0      | 0   | 0 | 0   | 0      | 0    |
| R/W:           | R  | R/W | R/W    | R/W  | R  | R/W | R/W    | R/W  | R | R/W | R/W    | R/W | R | R/W | R/W    | R/W  |

| Bit      | Bit Name | Initial<br>Value | R/W | Description   |
|----------|----------|------------------|-----|---|
| 15       | _        | 0                | R   | Reserved  |
|          |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 14 to 12 | IOBDn3   | 000              | R/W | I/O Control B   |
|          | [2:0]    |                  |     | When GRDn3 functions as the output compare register               |
|          |          |                  |     | 000: Compare match is not performed                               |
|          |          |                  |     | 001: Output level on compare match is 0                           |
|          |          |                  |     | 010: Output level on compare match is 1                           |
|          |          |                  |     | 011: Output level on compare match is toggled                     |
|          |          |                  |     | When GRDn3 functions as the capture register                      |
|          |          |                  |     | 100: Input capture is not performed                               |
|          |          |                  |     | 101: TCNT2Dn is captured on compare match A                       |
|          |          |                  |     | 110: Setting prohibited   |
|          |          |                  |     | 111: Setting prohibited   |
| 11       | _        | 0                | R   | Reserved  |
|          |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 10 to 8  | IOBDn2   | 000              | R/W | I/O Control B   |
|          | [2:0]    |                  |     | When GRDn2 functions as the output compare register               |
|          |          |                  |     | 000: Compare match is not performed                               |
|          |          |                  |     | 001: Output level on compare match is 0                           |
|          |          |                  |     | 010: Output level on compare match is 1                           |
|          |          |                  |     | 011: Output level on compare match is toggled                     |
|          |          |                  |     | When GRDn2 functions as the capture register                      |
|          |          |                  |     | 100: Input capture is not performed                               |
|          |          |                  |     | 101: TCNT2Dn is captured on compare match A                       |
|          |          |                  |     | 110: Setting prohibited   |
|          |          |                  |     | 111: Setting prohibited   |
|          |          |                  |     |   |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7      | _        | 0                | R   | Reserved  |
|        |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 6 to 4 | IOBDn1   | 000              | R/W | I/O Control B   |
|        | [2:0]    |                  |     | When GRDn1 functions as the output compare register               |
|        |          |                  |     | 000: Compare match is not performed                               |
|        |          |                  |     | 001: Output level on compare match is 0                           |
|        |          |                  |     | 010: Output level on compare match is 1                           |
|        |          |                  |     | 011: Output level on compare match is toggled                     |
|        |          |                  |     | When GRDn1 functions as the capture register                      |
|        |          |                  |     | 100: Input capture is not performed                               |
|        |          |                  |     | 101: TCNT2Dn is captured on compare match A                       |
|        |          |                  |     | 110: Setting prohibited   |
|        |          |                  |     | 111: Setting prohibited   |
| 3      | _        | 0                | R   | Reserved  |
|        |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 2 to 0 | IOBDn0   | 000              | R/W | I/O Control B   |
|        | [2:0]    |                  |     | When GRDn0 functions as the output compare register               |
|        |          |                  |     | 000: Compare match is not performed                               |
|        |          |                  |     | 001: Output level on compare match is 0                           |
|        |          |                  |     | 010: Output level on compare match is 1                           |
|        |          |                  |     | 011: Output level on compare match is toggled                     |
|        |          |                  |     | When GRDn0 functions as the capture register                      |
|        |          |                  |     | 100: Input capture is not performed                               |
|        |          |                  |     | 101: TCNT2Dn is captured on compare match A                       |
|        |          |                  |     | 110: Setting prohibited   |
| 1      |          |                  |     | 111: Setting prohibited   |

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

# 13.19.5 Down Counter Starting Registers D0 to D3 (DSTRD0 to DSTRD3)

DSTRD0 to DSTRD3 are 8-bit readable/writable registers that start the down counter. Setting the bits makes the down counter start.

DSTRD0 to DSTRD3 can be read from and written to in byte or word units.

DSTRD0 to DSTRD3 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3          | 2          | 1          | 0          |
|----------------|---|---|---|---|------------|------------|------------|------------|
|                | - | - | - | - | DSTD<br>n3 | DSTD<br>n2 | DSTD<br>n1 | DSTD<br>n0 |
| Initial value: | 0 | 0 | 0 | 0 | 0          | 0          | 0          | 0          |
| R/W:           | R | R | R | R | R/(W)*     | R/(W)*     | R/(W)*     | R/(W)*     |

<sup>\*</sup> Writing 0 is ignored. This bit is always read as 0 even if 1 is written.

| Bit    | Bit Name | Initial<br>Value | R/W    | Description   |
|--------|----------|------------------|--------|---|
| 7 to 4 | _        | All 0            | R      | Reserved  |
|        |          |                  |        | These bits are always read as 0. The write value should always be 0.  |
| 3      | DSTDn3   | 0                | R/(W)* | Down Counter Start Dnm  |
| 2      | DSTDn2   | 0                | R/(W)* | Setting these bits to 1 makes down counter Dn   |
| 1      | DSTDn1   | 0                | R/(W)* | - (DCNTDnm) start. The setting in these bits is always valid and regardless of the start trigger setting in the down  |
| 0      | DSTDn0   | 0                | R/(W)* | counter control register. When compare match B and writing 1 to these bits occurs at the same time if the down counter is set so that it is stopped on compare match B, compare match B takes priority and the down counter is not started. |
|        |          |                  |        | When $DCNTDn = H'000000$ , writing 1 to these bits has no effect.   |
|        |          |                  |        | 0: No operation   |
|        |          |                  |        | 1: Down counters (DCNTDn0 to DCNTDn3) are started   |

# 13.19.6 Down Counter Status Registers D0 to D3 (DSRD0 to DSRD3)

DSRD0 to DSRD3 is an 8-bit readable/writable register that indicates the state of the timer down counter (DCNTDnm).

DSRD0 to DSRD3 can be read from and written to in byte or word units.

DSRD0 to DSRD3 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3          | 2          | 1          | 0          |
|----------------|---|---|---|---|------------|------------|------------|------------|
|                | - | - | 1 | - | DSFD<br>n3 | DSFD<br>n2 | DSFD<br>n1 | DSFD<br>n0 |
| Initial value: | 0 | 0 | 0 | 0 | 0          | 0          | 0          | 0          |
| R/W:           | R | R | R | R | R*         | R*         | R*         | R*         |

Note: n = 0 to 3 (correspond to subblocks D0 to D3) \* DSDFDnm is a read-only bit and cannot be modified.

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 4 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 3   | DSFDn3   | 0                | R*  | Down Counter Status Flag Dnm   |
| 2   | DSFDn2   | 0                | R*  | These bits indicate enabling/disabling of down counter   |
| 1   | DSFDn1   | 0                | R*  | Dnm (DCNTDnm). When these bits are read as 1, the counter operation is enabled. If the TDE bit of ATUENR is  |
| 0   | DSFDn0   | 0                | R*  | 1 and the STRDn bit of TSTRD is 1, the counter is running. When these bits are read as 0, the counter operation is disabled, so counting cannot be in progress.  |
|     |          |                  |     | 0: Down counter is disabled  |
|     |          |                  |     | [Clearing conditions]  |
|     |          |                  |     | When the down counter is stopped by underflow  |
|     |          |                  |     | <ul> <li>When the condition to stop the down counter (compare<br/>match B)</li> </ul>  |
|     |          |                  |     | 1: Down counter is enabled   |
|     |          |                  |     | [Setting conditions]   |
|     |          |                  |     | <ul> <li>When writing 1 to the down counter start bit in the<br/>down counter starting register (DSTRDn)</li> </ul>  |
|     |          |                  |     | <ul> <li>When the condition set as the down counter start<br/>trigger is satisfied (compare match A or B)</li> </ul>   |
|     |          |                  |     | These flags are set or cleared regardless of the settings of the TDE bit in ATUENR and the STRDn bit in TSTRD.  Accordingly, if the TDE bit and the STRDn bit are not set to enable counting, the down counter is not actually running even if these bits indicate that counting is enabled. |

# 13.19.7 Down Counter Control Registers D0 to D3 (DCRD0 to DCRD3)

DCRD0 to DCRD3 are 16-bit readable/writable registers that starts the timer down counter (DCNTDnm). Starting and stopping by compare match A or B can be set. To change the TRGSELDn bit, stop the counter. Otherwise operation cannot be guaranteed.

The trigger source to start and stop the counter can be selected by the TRGSELD bits. The counter can be started by compare match A or B and can be stopped by compare match B.

The counter can also be started by writing 1 to the down counter start bit. The trigger source is always valid regardless of the TRGSELD bits and takes priority over other sources.

DCRD0 to DCRD3 can be read from and written to in byte- or word-units.

DCRD0 to DCRD3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

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

| Bit      | Bit Name  | Initial<br>Value | R/W | Description   |                         |  |  |  |  |
|----------|-----------|------------------|-----|---|-------------------------|--|--|--|--|
| 15       | _         | 0                | R   | Reserved  |                         |  |  |  |  |
|          |           |                  |     | This bit is always read as 0. The write value shound always be 0. |                         |  |  |  |  |
| 14 to 12 | TRGSELDn3 | 000              | R/W | Down Counter Start/Stop   | Trigger Select Dn3      |  |  |  |  |
|          | [2:0]     |                  |     | [Counter start trigger]   | [Counter stop trigger]  |  |  |  |  |
|          |           |                  |     | 000: No trigger   | 000: No trigger         |  |  |  |  |
|          |           |                  |     | 001: No trigger   | 001: Compare match B    |  |  |  |  |
|          |           |                  |     | 010: Compare match A  | 010: No trigger         |  |  |  |  |
|          |           |                  |     | 011: Compare match A  | 011: Compare match B    |  |  |  |  |
|          |           |                  |     | 100: Compare match B  | 100: No trigger         |  |  |  |  |
|          |           |                  |     | 101: Setting prohibited   | 101: Setting prohibited |  |  |  |  |
|          |           |                  |     | 110: Setting prohibited   | 110: Setting prohibited |  |  |  |  |
|          |           |                  |     | 111: Setting prohibited   | 111: Setting prohibited |  |  |  |  |

| Bit     | Bit Name  | Initial<br>Value | R/W | Description                             |                           |
|---------|-----------|------------------|-----|---|---------------------------|
| 11      | _         | 0                | R   | Reserved                                |                           |
|         |           |                  |     | This bit is always read as always be 0. | 0. The write value should |
| 10 to 8 | TRGSELDn2 | 000              | R/W | Down Counter Start/Stop                 | Trigger Select Dn2        |
|         | [2:0]     |                  |     | [Counter start trigger]                 | [Counter stop trigger]    |
|         |           |                  |     | 000: No trigger                         | 000: No trigger           |
|         |           |                  |     | 001: No trigger                         | 001: Compare match B      |
|         |           |                  |     | 010: Compare match A                    | 010: No trigger           |
|         |           |                  |     | 011: Compare match A                    | 011: Compare match B      |
|         |           |                  |     | 100: Compare match B                    | 100: No trigger           |
|         |           |                  |     | 101: Setting prohibited                 | 101: Setting prohibited   |
|         |           |                  |     | 110: Setting prohibited                 | 110: Setting prohibited   |
|         |           |                  |     | 111: Setting prohibited                 | 111: Setting prohibited   |
| 7       | _         | 0                | R   | Reserved                                |                           |
|         |           |                  |     | This bit is always read as always be 0. | 0. The write value should |
| 6 to 4  | TRGSELDn1 | 000              | R/W | Down Counter Start/Stop                 | Trigger Select Dn1        |
|         | [2:0]     |                  |     | [Counter start trigger]                 | [Counter stop trigger]    |
|         |           |                  |     | 000: No trigger                         | 000: No trigger           |
|         |           |                  |     | 001: No trigger                         | 001: Compare match B      |
|         |           |                  |     | 010: Compare match A                    | 010: No trigger           |
|         |           |                  |     | 011: Compare match A                    | 011: Compare match B      |
|         |           |                  |     | 100: Compare match B                    | 100: No trigger           |
|         |           |                  |     | 101: Setting prohibited                 | 101: Setting prohibited   |
|         |           |                  |     | 110: Setting prohibited                 | 110: Setting prohibited   |
|         |           |                  |     | 111: Setting prohibited                 | 111: Setting prohibited   |
| 3       | _         | 0                | R   | Reserved                                |                           |
|         |           |                  |     | This bit is always read as always be 0. | 0. The write value should |

| Bit    | Bit Name  | Initial<br>Value | R/W | Description             |                         |
|--------|-----------|------------------|-----|-------------------------|-------------------------|
| 2 to 0 | TRGSELDn0 | 000              | R/W | Down Counter Start/Stop | Trigger Select Dn0      |
|        | [2:0]     |                  |     | [Counter start trigger] | [Counter stop trigger]  |
|        |           |                  |     | 000: No trigger         | 000: No trigger         |
|        |           |                  |     | 001: No trigger         | 001: Compare match B    |
|        |           |                  |     | 010: Compare match A    | 010: No trigger         |
|        |           |                  |     | 011: Compare match A    | 011: Compare match B    |
|        |           |                  |     | 100: Compare match B    | 100: No trigger         |
|        |           |                  |     | 101: Setting prohibited | 101: Setting prohibited |
|        |           |                  |     | 110: Setting prohibited | 110: Setting prohibited |
|        |           |                  |     | 111: Setting prohibited | 111: Setting prohibited |

Note: n = 0 to 3 (n denotes subblock number)

## 13.19.8 Timer Status Registers D0 to D3 (TSRD0 to TSRD3)

TSRD0 to TSRD3 are 16-bit readable/writable registers that control overflow on timer counters (TCNT2Dn and TCNT1Dn) in subblocks D0 to D3, underflow on timer down counters (DCNTDnm) in channels 0 to 3, and compare match on output compare register (OCRDnm) and general register (GRDnm)

Interrupt requests can be issued by using overflow flags 2 and 1, underflow flag, and compare match flags A and B. To issue interrupt requests, set the corresponding bit in timer interrupt enable register (TIERDn). By using compare matches A and B, pulses can be output to the A/D converter to start conversion. Eight pulse output lines are available in subblocks D0 and D1.

TSRD0 to TSRD3 can be read from and written to in byte- or word-units.

TSRD0 to TSRD3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 15 | 14 | 13         | 12         | 11         | 10         | 9          | 8          | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|----------------|----|----|------------|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | -  | -  | OVF<br>2Dn | OVF<br>1Dn | UDF<br>Dn3 | UDF<br>Dn2 | UDF<br>Dn1 | UDF<br>Dn0 | CMF<br>ADn3 | CMF<br>ADn2 | CMF<br>ADn1 | CMF<br>ADn0 | CMF<br>BDn3 | CMF<br>BDn2 | CMF<br>BDn1 | CMF<br>BDn0 |
| Initial value: | 0  | 0  | 0          | 0          | 0          | 0          | 0          | 0          | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R  | R  | R/(W)*      | R/(W)*      | R/(W)*      | R/(W)*      | R/(W)*      | R/(W)*      | R/(W)*      | R/(W)*      |

<sup>\*</sup> Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

| Bit    | Bit Name | Initial<br>Value | R/W    | Description   |
|--------|----------|------------------|--------|---|
| 15, 14 | _        | All 0            | R      | Reserved  |
|        |          |                  |        | These bits are always read as 0. The write value should always be 0.  |
| 13     | OVF2Dn   | 0                | R/(W)* | Overflow Flag 2Dn   |
|        |          |                  |        | Indicates whether or not timer counter 2Dn (TCNT2Dn) has overflowed. This flag cannot be set to 1 by software.  |
|        |          |                  |        | This bit is set to 1 when TCNT2Dn is incremented while it is H'FFFFFF. Writing H'000000 to TCNT2Dn or starting TCNT2Dn from a initial value of H'000000 has no effect on this bit.  |
|        |          |                  |        | When writing to TCNT2Dn at the same time as incrementation while it is H'FFFFFF, this bit is set to 1. However, TCNT2Dn is started from the written value.  |
|        |          |                  |        | When an assertion of counter clearing signal from timer B and overflow occur, overflow does not occur. The overflow can be notified as an overflow interrupt request by setting the overflow enable flag (OVE2Dn) in timer interrupt enable register Dn (TIERDn). |
|        |          |                  |        | 0: TCNT2Dn has not overflowed   |
|        |          |                  |        | [Clearing condition]  |
|        |          |                  |        | When writing 0 to this bit after reading it as 1  |
|        |          |                  |        | 1: TCNT2Dn has overflowed   |
|        |          |                  |        | [Setting condition]   |
|        |          |                  |        | <ul> <li>When TCNT2Dn overflowed (from H'FFFF FF to H'0000<br/>00)</li> </ul>   |

|     |          | Initial |        |   |
|-----|----------|---------|--------|---|
| Bit | Bit Name | Value   | R/W    | Description   |
| 12  | OVF1Dn   | 0       | R/(W)* | Overflow Flag 1Dn   |
|     |          |         |        | Indicates whether or not timer counter 1Dn (TCNT1Dn) has overflowed. This flag cannot be set to 1 by software.  |
|     |          |         |        | This bit is set to 1 when TCNT1Dn is incremented while it is H'FFFFFF. Writing H'000000 to TCNT1Dn or starting TCNT1Dn from a initial value of H'000000 has no effect on this bit.  |
|     |          |         |        | When writing to TCNT1Dn at the same time as incrementation while it is H'FFFFFF, this bit is set to 1. However, TCNT1Dn is started from the written value.  |
|     |          |         |        | When an assertion of counter clearing signal from timer B and overflow occur, overflow does not occur. The overflow can be notified as an overflow interrupt request by setting the overflow enable flag (OVE1Dn) in timer interrupt enable register Dn (TIERDn). |
|     |          |         |        | 0: TCNT1Dn has not overflowed   |
|     |          |         |        | [Clearing condition]  |
|     |          |         |        | When writing 0 to this bit after reading it as 1  |
|     |          |         |        | 1: TCNT1Dn has overflowed   |
|     |          |         |        | [Setting condition]   |
|     |          |         |        | When TCNT1Dn overflowed (from H'FFFF FF to H'0000 00)   |

| Bit | Bit Name | Value |        |   |
|-----|----------|-------|--------|---|
|     |          |       | R/W    | Description   |
| 11  | UDFDn3   | 0     | R/(W)* | Underflow Flag Dnm  |
| 10  | UDFDn2   | 0     | R/(W)* | These bits indicate whether or not the timer down counter   |
| 9   | UDFDn1   | 0     | R/(W)* | (DCNTDnm) has underflowed.  |
| 8   | UDFDn0   | 0     | R/(W)* | This bit is set to 1 when DCNTDnm is to be decremented while it is H'000000. DCNTDnm holds H'000000 on underflow. Writing H'FFFFFF to DCNTDnm has no effect on these bits. These flags cannot be set to 1 by software.                        |
|     |          |       |        | These bits in subblocks D0 to D3 (16 channels) can be cleared by the ACK signal from the DMAC.  |
|     |          |       |        | These bits are initialized to 0 by a reset. Although DCNTDnm is initialized to H'000000, these flags do not indicate underflow because DCNTDnm has not started. For details on control of DCNTDnm, see descriptions of timer D down counters. |
|     |          |       |        | 0: DCNTDnm has not underflowed  |
|     |          |       |        | [Clearing conditions]   |
|     |          |       |        | When writing 0 to these bits after reading them as 1  |
|     |          |       |        | <ul> <li>When DCNTDnm is cleared by an assertion of the ACK<br/>signal from the DMAC (supported only by subblocks 0 to<br/>3)</li> </ul>  |
|     |          |       |        | 1: DCNTDnm has underflowed  |
|     |          |       |        | [Setting condition]   |
|     |          |       |        | When DCNTDnm underflowed (decremented while<br>DCNTDnm = H'0000 00)   |

|     |          | Initial |        |  |
|-----|----------|---------|--------|--|
| Bit | Bit Name | Value   | R/W    | Description  |
| 7   | CMFADn3  | 0       | R/(W)* | Compare Match A Flag Dnm   |
| 6   | CMFADn2  | 0       | R/(W)* | These bits indicate whether or not compare match between   |
| 5   | CMFADn1  | 0       | R/(W)* | the output compare register (OCRDnm) and TCNT1Dn has cocurred. This flag cannot be set to 1 by software.   |
| 4   | CMFADn0  | 0       | R/(W)* | When operation of compare match between OCRDnm and TCNT1Dn is enabled by the setting in timer I/O control register 1 (TIOR1Dn), compare match operation is performed regardless of the state of TCNT1Dn. These bits are set to 1 on the first edge of the P $\varphi$ clock after the values in TCNT1Dn and OCRDnm match,  |
|     |          |         |        | Even if these compare match flags are cleared to 0 by software while TCNT1Dn = OCRDnm after the compare match is detected, these bits are not set to 1 again.  |
|     |          |         |        | A single pulse, signaling detection of compare match A, whose width is equal to the cycle of the P $\phi$ clock is output to activate the A/D converter (supported by 8 channels in subblocks D0 and D1).  |
|     |          |         |        | To clear these bits, write 0 to these bits after reading them as 1.  |
|     |          |         |        | If TCNT1Dn matches OCRDnm again before the status flag is cleared, the compare match A is detected and the status flag is rewritten with 1.  |
|     |          |         |        | An interrupt request indicating compare match A can be output by setting the compare match A enable bit (CMEADn) in timer interrupt enable register Dn (TIERDn) to 1. The signal line is shared with the compare match B flag. The interrupt signal is asserted either when the compare match A flag and compare match A enable bit are both set to 1 or when the compare match B flag and compare match B enable bit are both set to 1. |
|     |          |         |        | 0: Compare match A has not occurred  |
|     |          |         |        | [Clearing condition]   |
|     |          |         |        | When writing 0 to these bits after reading them as 1   |
|     |          |         |        | 1: Compare match A has occurred  |
|     |          |         |        | [Setting condition]  |
|     |          |         |        | <ul> <li>When the values in timer counter 1 (TCNT1Dn) and the<br/>output compare register (OCRDnm) match while<br/>operation of compare match with OCRDnm is enabled</li> </ul>  |

|     |          | Initial          |        |  |
|-----|----------|------------------|--------|--|
| Bit | Bit Name | Initial<br>Value | R/W    | Description  |
| 3   | CMFBDn3  | 0                | R/(W)* | Compare Match B Flag Dnm   |
| 2   | CMFBDn2  | 0                | R/(W)* | These bits indicate whether or not compare match between   |
| 1   | CMFBDn1  | 0                | R/(W)* | the general register (GRDnm) and TCNT2Dn has occurred. This flag cannot be set to 1 by software.   |
| 0   | CMFBDn0  | 0                | R/(W)* | When operation of compare match between GRDnm and TCNT2Dn is enabled by the setting in timer I/O control register 2 (TIOR2Dn), compare match operation is performed regardless of the state of TCNT2Dn. These bits are set to 1 on the first edge of the P $_{\varphi}$ clock after the values in TCNT2Dn and GRDnm match.   |
|     |          |                  |        | Even if these compare match flags are cleared to 0 by software while TCNT2Dn = GRDnm after the compare match is detected, these bits are not set to 1 again.   |
|     |          |                  |        | A single pulse, signaling detection of compare match B, whose width is equal to the cycle of the P $\phi$ clock is output to activate the A/D converter (supported by 8 channels in subblocks D0 and D1).  |
|     |          |                  |        | To clear these bits, write 0 to these bits after reading them as 1.  |
|     |          |                  |        | If TCNT2Dn matches GRDnm again before the status flag is cleared, the compare match B is detected and the status flag is rewritten with 1.   |
|     |          |                  |        | An interrupt request indicating compare match B can be output by setting the compare match B enable bit (CMEBDn) in timer interrupt enable register Dn (TIERDn) to 1. The signal line is shared with the compare match A flag. The interrupt signal is asserted either when the compare match A flag and compare match A enable bit are both set to 1 or when the compare match B flag and compare match B enable bit are both set to 1. |
|     |          |                  |        | 0: Compare match B has not occurred  |
|     |          |                  |        | [Clearing condition]   |
|     |          |                  |        | When writing 0 to these bits after reading them as 1   |
|     |          |                  |        | 1: Compare match B has occurred  |
|     |          |                  |        | [Setting condition]  |
|     |          |                  |        | When the values in timer counter 2 (TCNT2Dn) and   |
|     |          |                  |        | ara a sa li sa ariada si (CDD) a sa li sa a li sa li la la a sa a : ti - : f   |

Notes: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number).

general register (GRDnm) match while operation of

compare match with GRDnm is enabled

\* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

### 13.19.9 Timer Interrupt Enable Registers D0 to D3 (TIERD0 to TIERD3)

TIERD0 to TIERD3 are 16-bit readable/writable registers that enable and disable interrupt requests for overflow on two timer counters (TCNT2Dn and TCNT1Dn) and compare matches between TCNT1Dn and the output compare register (OCRDnm) and between TCNT2Dn and general register (GRDnm).

TIERD0 to TIERD3 can be read from and written to in byte- or word-units.

TIERD0 to TIERD3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 15 | 14 | 13         | 12         | 11         | 10         | 9          | 8          | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|----------------|----|----|------------|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | -  | -  | OVE2<br>Dn | OVE1<br>Dn | UDE<br>Dn3 | UDE<br>Dn2 | UDE<br>Dn1 | UDE<br>Dn0 | CME<br>ADn3 | CME<br>ADn2 | CME<br>ADn1 | CME<br>ADn0 | CME<br>BDn3 | CME<br>BDn2 | CME<br>BDn1 | CME<br>BDn0 |
| Initial value: | 0  | 0  | 0          | 0          | 0          | 0          | 0          | 0          | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R  | R  | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 15, 14 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 13     | OVE2Dn   | 0                | R/W | Overflow Interrupt Enable 2Dn  |
|        |          |                  |     | Enables and disables an interrupt request when timer counter 2Dn (TCNT2Dn) has overflowed. By setting this bit, overflow flag 2 (OVF2Dn) in timer status register (TSRDn) can be used as a interrupt request source. |
|        |          |                  |     | 0: Interrupt request by OVF2Dn is disabled   |
|        |          |                  |     | 1: Interrupt request by OVF2Dn is enabled  |
| 12     | OVE1Dn   | 0                | R/W | Overflow Interrupt Enable 1Dn  |
|        |          |                  |     | Enables and disables an interrupt request when timer counter 1Dn (TCNT1Dn) has overflowed. By setting this bit, overflow flag 1 (OVF1Dn) in timer status register (TSRDn) can be used as a interrupt request source. |
|        |          |                  |     | 0: Interrupt request by OVF1D is disabled  |
| -      |          |                  |     | 1: Interrupt request by OVF1D is enabled   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 11  | UDEDn3   | 0                | R/W | Underflow Interrupt Enable Dnm   |
| 10  | UDEDn2   | 0                | R/W | These bits enable and disable an interrupt request when timer down counter Dnm (DCNTDnm) has underflowed. By setting these bits, underflow flag (UDFDnm) in timer status register (TSRDn) can be used as a interrupt request source.   |
| 9   | UDEDn1   | 0                | R/W |  |
| 8   | UDEDn0   | 0                | R/W |  |
|     |          |                  |     | 0: Interrupt request by UDFDnm is disabled   |
|     |          |                  |     | 1: Interrupt request by UDFDnm is enabled  |
| 7   | CMEADn3  | 0                | R/W | Compare Match Interrupt Enable Dnm   |
| 6   | CMEADn2  | 0                | R/W | These bits enable and disable an interrupt request when compare match between output compare register (OCRDnm) and TCNT1Dn has occurred. By setting these bits, compare match A flag (CMFADn) in timer status register (TSRDn) can be used as a interrupt request source. However, the signal line is shared with compare match B. |
| 5   | CMEADn1  | 0                | R/W |  |
| 4   | CMEADn0  | 0                | R/W |  |
|     |          |                  |     | 0: Interrupt request by CMFADnm is disabled  |
|     |          |                  |     | 1: Interrupt request by CMFADnm is enabled   |
| 3   | CMEBDn3  | 0                | R/W | Compare Match Interrupt Enable Dnm   |
| 2   | CMEBDn2  | 0                | R/W | These bits enable and disable an interrupt request when compare match between output compare register (GRDnm) and TCNT2Dn has occurred. By setting these bits, compare match B flag (CMFBDn) in timer status register (TSRDn) can be used as a interrupt request source. However, the signal line is shared with compare match A.  |
| 1   | CMEBDn1  | 0                | R/W |  |
| 0   | CMEBDn0  | 0                | R/W |  |
|     |          |                  |     | 0: Interrupt request by CMFBDnm is disabled  |
|     |          |                  |     | 1: Interrupt request by CMFBDnm is enabled   |

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number).

# 13.19.10 Compare Match Pulse Output Control Registers D0 and D1 (CMPOD0 and CMPOD1)

CMPOD0 and CMPOD1 are 8-bit readable/writable registers that select whether or not a pulse to activate the A/D converter is output on compare matches A and B.

CMPOD0 and CMPOD1 can be read from and written to in byte or word units.

CMPOD0 and CMPOD1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           |             |             |             |             | 3           |             |             |             |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | CMP<br>BDn3 | CMP<br>BDn2 | CMP<br>BDn1 | CMP<br>BDn0 | CMP<br>ADn3 | CMP<br>ADn2 | CMP<br>ADn1 | CMP<br>ADn0 |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |

Note: n = 0, 1 (correspond to subblocks D0 and D1)

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 7   | CMPBDn3  | 0                | R/W | Compare Match B Pulse Output Control   |
| 6   | CMPBDn2  | 0                | R/W | These bits select whether or not a pulse is output on  |
| 5   | CMPBDn1  | 0                | R/W | compare match B. When these bits are set to 1, an active—low pulse whose width is equal to the cycle of the P∳ clock |
| 4   | CMPBDn0  | 0                | R/W | is output on compare match B.  |
|     |          |                  |     | 0: Compare match B pulse is not output   |
|     |          |                  |     | 1: Compare match B pulse is output (pulse width = cycle of the $P\phi$ clock)  |
| 3   | CMPADn3  | 0                | R/W | Compare Match A Pulse Output Control   |
| 2   | CMPADn2  | 0                | R/W | These bits select whether or not a pulse is output on  |
| 1   | CMPADn1  | 0                | R/W | compare match A. When these bits are set to 1, an active—low pulse whose width is equal to the cycle of the P∳ clock |
| 0   | CMPADn0  | 0                | R/W | is output on compare match A.  |
|     |          |                  |     | 0: Compare match A pulse is not output   |
|     |          |                  |     | 1: Compare match A pulse is output (pulse width = cycle of the $P\phi$ clock)  |

Note: n = 0 to 3 (n denotes subblock number)

#### 13.19.11 Timer Output Control Registers D0 to D3 (TOCRD0 to TOCRD3)

TOCRD0 to TOCRD3 are 8-bit readable/writable registers that select whether or not signals on output pins for subblock Dn (TODnmA and TODnmB) are inverted.

Signals on pins TODnmA and TODnmB are inverted on the first edge of the  $P\phi$  clock after the output inversion select Dn bit is set to 1. This function is not affected by the operating state of timer counters 1Dn and 2Dn (TCNT1Dn, TCNT2Dn).

The TONEBDn bit controls four outputs (TODn0B, TODn1B, TODn2B, TODn3B) in a single subblock. The TONEADn bit controls four outputs (TODn0A, TODn1A, TODn2A, TODn3A) in a single subblock. These bits cannot control individual signals independently (channel control is not available).

Output levels on pins TODnmA and TODnmB are initialized to a level of 0 (when TONEADn = 0 and TONEBDn = 0).

TOCRD0 to TOCRD3 can be read from and written to in byte or word units.

TOCRD0 to TOCRD3 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1           | 0           |
|----------------|---|---|---|---|---|---|-------------|-------------|
|                | 1 | - | - | - | - | - | TONE<br>BDn | TONE<br>ADn |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0           | 0           |
| R/W:           | R | R | R | R | R | R | R/W         | R/W         |

Note: n = 0 to 3 (correspond to subblocks D0 to D3)

|        |          | Initial |     |  |
|--------|----------|---------|-----|--|
| Bit    | Bit Name | Value   | R/W | Description  |
| 7 to 2 | _        | All 0   | R   | Reserved   |
|        |          |         |     | These bits are always read as 0. The write value should always be 0. |
| 1      | TONEBDn  | 0       | R/W | Output Inversion Select TODnmB                                       |
|        |          |         |     | Selects whether or not the output level on pin TODnmB is inverted    |
|        |          |         |     | 0: Output level is not inverted                                      |
|        |          |         |     | 1: Output level is inverted  |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 0   | TONEADn  | 0                | R/W | Output Inversion Select TODnmA                                    |
|     |          |                  |     | Selects whether or not the output level on pin TODnmA is inverted |
|     |          |                  |     | 0: Output level is not inverted                                   |
|     |          |                  |     | 1: Output level is inverted                                       |

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

## 13.19.12 Timer Offset Base Registers D0 to D3 (OSBRD0 to OSBRD3)

OSBRD0 to OSBRD3 are 32-bit read-only registers that are used only for capture. The value in timer counter 1Dn (TCNT1Dn) is captured in these registers by a trigger signal from timer A. Pin TIA01 or TIA02 can be selected as a trigger. Trigger signals for subblocks D0 to D2 are selected by bit EVOSEL2A in TCRA and trigger signals for subblock D3 are selected by bit EVOSEL2B in TCRA. For details, see section 13.10.1, Timer Control Register A (TCRA).

OSBRD0 to OSBRD3 can be read from in longword units.

OSBRD0 to OSBRD3 are initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |
| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|                |    |    |    |    |    |    |    |    | -  | -  | -  | -  | -  | -  | -  | -  |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

# **13.19.13** Timer Counter 1D0 to 1D3 (TCNT1D0 to TCNT1D3)

TCNT1D0 to TCNT1D3 are 32-bit readable/writable registers driven by the clock selected in the CKSEL1Dn[2:0] bits in timer control register Dn (TCRDn). These counters are started by setting the bit in timer start register (TSTRD) to 1.

When the counter overflows, the over flag (OVF1Dn) in timer status register Dn (TSRDn) is set to 1.

TCNT1D0 to TCNT1D3 can be read from and written to in longword units.

TCNT1D0 to TCNT1D3 is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

# 13.19.14 Timer Counters 2D0 to 2D3 (TCNT2D0 to TCNT2D3)

TCNT2D0 to TCNT2D3 are 32-bit readable/writable registers driven by the clock selected in the CKSEL2Dn[2:0] bits in timer control register Dn (TCRDn). These counters are started by setting the bit in the timer start register Dn (TSTRD) to 1.

When the counter overflows, the over flag (OVF2Dn) in timer status register Dn (TSRDn) is set to 1.

TCNT2D0 to TCNT2D3 can be read from and written to in longword units.

TCNT2D0 to TCNT2D3 is initialized to H'000000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|                |     |     |     |     |     |     |     |     | _   | _   | -   | -   | -   | _   | -   | -   |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

## 13.19.15 Output Compare Registers D00 to D33 (OCRD00 to OCRD33)

OCRD00 to OCRD33 are 32-bit readable/writable registers. The upper 24 bits are available.

OCRDnm is constantly compared with the up counter TCNT1Dn. When compare match operation is selected by bit IOADnm in TIOR1Dn, the compare match A flag (CMFADnm) in TSRDn is set to 1 on the first edge of the  $P\phi$  clock after the values in TCNT1Dn and OCRDnm match. When compare match A is selected by an output source select bit, a signal is output to pin TODnmA on compare match.

When compare match A is selected as a down counter starting trigger by the TRGSELDnm bit in DCRDnm, DCNTDnm is ready to be counted down on compare match A.

When the down counter (DCNTDnm) is ready, it is started in synchronization with the down counter clock. At this time, a one-shot pulse can be output on pin TODnmB. If compare match A and down-counter stop trigger are output at the same time, output is disabled without any pulse.

When TCNT1Dn overflows to change from H'FFFFFF to H'000000 and OCRDnm is set to H'000000, compare match is detected.

By setting bit CMEADn in timer interrupt enable register Dn (TIERDn) to 1, an interrupt request can be issued on compare match A. (The output line is shared with compare match B. Confirm that which interrupt is occurred by TSRDn).

OCRD00 to OCRD33 can be read from and written to in longword units.

OCRD00 to OCRD33 are initialized to H'FFFFFF00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

#### 13.19.16 General Registers D00 to D33 (GRD00 to GRD33)

GRD00 to GRD33 are 32-bit readable/writable registers that function as the capture register and compare match register. The functions are switched by timer I/O control register 2Dn (TIOR2Dn).

For the capture function, these registers capture the value in TCNT2Dn on compare match A. Even if TCNT2Dn is stopped (the TDE bit in ATUENR is cleared to 0 or the STRDn bit in TSTRD is cleared to 0), capture continues and the value in TCNT2Dn stopped is captured by GRDnm.

For the compare match function, these registers are constantly compared with TCNT2Dn. When compare match is enabled by the IOBDnm bit in TIOR2Dn, the CMFBDnm bit in TSRDn is set to 1 in synchronization with the  $P\phi$  clock after the values in TCNT2Dn and GRDnm match. When compare match B is selected as the output source by the OSSDnm bit, a signal is output on pin TODnmA on compare match.

When compare match B is selected as the down counter starting trigger by the TRGSELDnm bit in DCRDnm, DCNTDnm is ready to be counted down on compare match B.

When DCNTDnm is ready, it is decremented in synchronization with the down counter clock. At this time, a one-shot pulse can be output on pin TODnmB.

When compare match B is selected as the down counter stopping trigger by the TRGSELDnm bit in DCRDnm, DCNTDnm is not ready for counting down. The counter is cleared to 0 and the output signal on pin TODnmB is negated (output of one-shot pulse is terminated) in synchronization with the down counter clock after it is not ready because of compare match B.

When TCNT2Dn overflows to change from H'FFFFFF to H'000000 and GRDnm is set to H'000000, compare match is detected.

By setting bit CMEBDn in timer interrupt enable register Dn (TIERDn) to 1, an interrupt request can be issued on compare match B. (The output line is shared with compare match A. Confirm that which interrupt is occurred by TSRDn).

GRD00 to GRD33 can be read from and written to in longword units.

GRD00 to GRD33 are initialized to H'FFFF FF00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

#### 13.19.17 Timer Down Counters D00 to 33 (DCNTD00 to DCNTD33)

DCNTD00 to DCNTD33 are 32-bit readable/writable registers driven by the clock selected in the DCSELDn[2:0] bit in timer control register Dn (TCRDn).

DCNTDnm is controlled by the down counter control register Dn (DCRDn). This register starts to be decremented; when compare match A or B is detected; when 1 is written to the DSTDnm bit in DSTRDn. Counting is stopped when DCNTDnm underflows or on compare match B. While the down counter is enabled, it is decremented every input of the down counter clock.

Decrementation is enabled; on the first edge of the  $P\phi$  clock (the same as compare match A) after the values in TCNT1Dn and OCRDnm match; on the first edge of the  $P\phi$  clock (the same as compare match B) after the values in TCNT2Dn and GRDnm match; on the first edge of the  $P\phi$  clock after the DSTDnm bit is set to 1. Decrementation is enabled until DCNTDnm underflows or until the first edge of the  $P\phi$  clock (the same as compare match B) after the values in TCNT2Dn and GRDnm match. The down counter is decremented every input of the down counter clock while it is enabled.

Once DCNTDnm is enabled, it remains enabled until DCNTDnm underflows or on compare match B (when counter stopping trigger is selected). While it is enabled, another counter starting trigger or writing 1 to DSTDnm bit has no effect on the enabled state of the counter.

The counter is stopped on the first edge of the down counter clock after compare match B if the counter stopping trigger is selected and then is cleared to H'000000. When the counter is started or stopped by compare match A and writing 1 to DSTDnm simultaneously, counter stopping trigger takes priority. The counter is not decremented and no signal on pin TODnmB is output. Moreover, when no down counter clock is input during the enabled state, the counter is not decremented. The counter which is stopped does not change the value. If a value other than H'000000 is set in DCNTDnm after decrementation is terminated by underflow, the counter is not decremented until the counter stopping source is activated.

DCNTD00 to DCNTD33 can be read from and written to in longword units.

DCNTD00 to DCNTD33 are initialized to H'000000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16            |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------------|
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |               |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0             |
| R/W:           | R/W           |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0             |
| Dit.           | 15  | 14  | 15  | 12  |     | 10  |     |     |     |     |     |     |     |     |     | $\overline{}$ |
|                |     |     |     |     |     |     |     |     | -   | -   | -   | -   | -   | -   | -   | -             |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0             |
| R/W:           | R/W | R   | R   | R   | R   | R   | R   | R   | R             |

# 13.20 Operations of Timer D

One-shot pulse can be output from timer D. By using compare match A or B as a start trigger of a down counter, one-shot pulse with an offset can also be output.

Setting the TDE bit in ATUENR and bit STRDn in timer start register (TSTRD) to 1 makes two up-counters (TCNT1Dn and TCNT2Dn) in subblock Dn start operation.

The down counter is started by setting bit DSTDnm in down counter starting register D (DSTRD) to 1, compare match A, or compare match B. Compare matches A and B are selected by bit TRGSELnm in down counter control register (DCRD). When any of these triggers occurs, DCNTDnm driven by the down counter clock is started.

The down counter is stopped on an underflow of the down counter and compare match B selected by bit TRFSELmn. The down counter is stopped immediately after it underflows (the counter is to be decremented when the value is H'000000). It is stopped and cleared to H'000000 on the first edge of the down counter clock after the trigger is detected.

For compare match A between TCNT1Dn and OCRDnm and compare match B between TCNT2Dn and GRDnm, the CMFADnm and CMFBDnm bits in TSRDn are set to 1 on the first edge of the  $P\phi$  clock after the compare match.

A signal on pin TODnmA is output when the output source selected in the OSSDnm bit in timer I/O control register 1D (TIOR1Dn) is activated. For example, assume that compare match A is selected. A signal level set by the IOAnm bit is output on pin TODnmA on the first edge of the  $P\phi$  clock after the compare match between TCNT1Dn and OCRDnm.

Output of the one-shot pulse is synchronized with the down counter clock in a way similar to the down counter operation. Since the three counter starting sources are synchronized with the  $P\phi$  clock, a signal on pin TODnmB is output on the first edge of the down counter clock after the source is activated. Underflow of the down counter, which is a counter stopping source, is synchronized with the down counter clock and compare match B is synchronized with the  $P\phi$  clock. As to negation timing, the TODnmB signal is negated in synchronization with underflow and on the first edge of the down counter clock after the compare match B.

**Table 13.14 Output Timing for One-Shot Pulse (TODnmB)** 

| C | output Pin | Assertion Timing   | Negation Timing  | Initial Value                    |
|---|------------|--|--|----------------------------------|
| Т | ODnmB      | On the first edge of the down counter clock after the counter starting source is activated | On the first edge of the<br>down counter clock after<br>compare match or on<br>DCNTDnm underflow | 0 (inverted depending on TOCRDn) |

Note: If an assertion and a negation occur simultaneously, the negation takes priority.

The initial values output on pins TODnmA and TODnmB are 0. However, the output level can be inverted by setting output control register Dn (TOCRDn). Setting bit TONEADn to 1 makes pin TODnmA in subblock Dn inverted and setting bit TONEBDn to 1 makes pin TODnmB in subblock Dn inverted.

An interrupt request can be output by setting bits CMEADnm and CMEBDnm to 1 using compare match A or B flag. The signal line is shared with compare matches A and B. When the status flag and enable bit for each interrupt are both set to 1, the output level of the interrupt request is asserted (active-low signal).

To set the clocks for TCNT1Dn, TCNT2Dn, or DCNTDnm or values in registers such as DCNTDnm, OCRDnm, and GRDnm while TCNT1Dn or TCNT2Dn is in operation, note that the value to be set may lead to malfunction. For example, while setting the compare match value, the counter value may exceed the value to be set.

Figure 13.28 shows an operation example of one-shot pulse output for channel 0 in subblock D0.

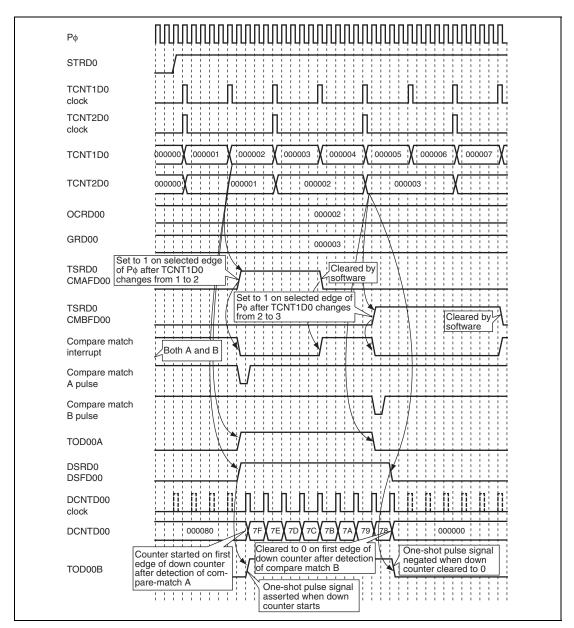


Figure 13.28 Operation Example of One-Shot Pulse Output (1) (Counting Started on Compare Match A and Stopped on Compare Match B)

Figure 13.28 shows operations of the down counter when the counter is started on compare match A and is stopped on compare match B. It also shows the assertion and negation of a one-shot pulse. Compare match A as the counter starting trigger and compare match B as the counter stopping trigger are set by the TRGSELD00 bits in DCRD0. Both matches are selected as the source of the output signal by the OSSD00 bits in TIOR1D0 and output levels are set by the IOAD00 and IOBD00 bits. A logical zero for compare match A and a logical one for compare match A are selected.

TCNT1D0 and TCNT2D0 are started for counting up on the first edge of the counter clock after the counter Dn start bit in timer start register (TSTRD) is set to 1. The compare match A status flag (CMFAD00) is set to 1 on the first edge of the  $P\phi$  clock after the values in TCNT1D0 and the output compare register (OCRD00) match. At the same time, a level of 1 is output on pin TOD00A and the down counter status flag (DSFD00) is set to 1 to make the down counter ready for counting down. The down counter keeps the ready state until compare match B or DCNTD00 underflow. DCNTD00 is started by the edge of the down counter clock. At this time, a level of 1 is output on pin TOD00B.

The compare match B flag (CMFBD00) is set to 1 on the first edge of the P\$\phi\$ clock after the values in GRD00 and TCNT2D0 match. At this time, a level of 0 is output on pin TOD00A. The down counter is cleared and a one-shot pulse (TOD00B) is terminated.

An interrupt request can be issued on compare matches A and B. For compare match A, set the CMEAD00 bit in timer interrupt enable register D0 (TIERED0) and for compare match B, set the CMEBD00 bit. Since the signal line is shared with both matches, read the CMAFD00 and CMBFD00 bits in TSRD0 to know the interrupt source.

Moreover, a pulse for A/D activation can be output by setting compare match pulse output control register D0 (CMPOD0) at the same time as the interrupt request to be output. The pulse width is equal to the cycle of the  $P\phi$  clock.

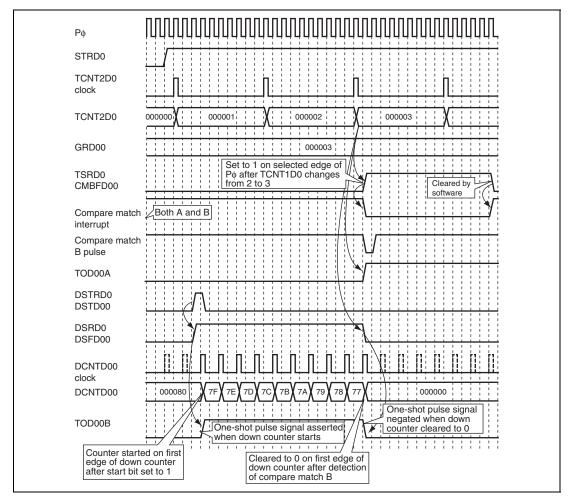


Figure 13.29 Operation Example of One-Shot Pulse Output (2) (Counting Started by Writing 1 to Counter Starting Bit and Stopped on Compare Match B)

Figure 13.29 shows an operation example when the down counter is started by writing 1 to the down counter starting bit. In this example, the counter starting trigger is not selected and compare match B is selected as the counter stopping trigger (TRGSELD00 in DCRD0). The source of the output signal is compare match B (OSSD00 in TIOR1D0) and logical one is output (IOBD00 in TIOR1D0).

Figure 13.30 shows an operation example of a one-shot pulse output for channel 0 in subblock D0. Setting the DSTD00 bit in the down counter starting register (DSTRD0) sets the down counter status flag (DSFD00) to 1. This makes the down counter ready for counting down. DCNTD00 is

started on the first edge of the down counter clock after DSFD00 is set to 1. At this time, a level of 1 is output on pin TOD00B.

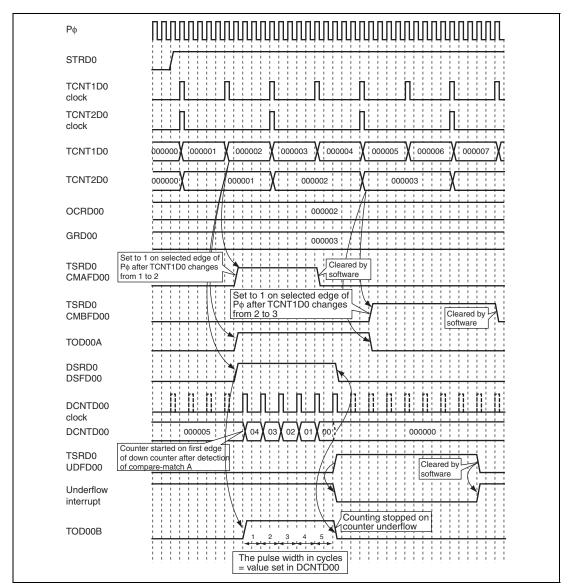


Figure 13.30 Operation Example of One-Shot Pulse Output (3) (Underflow Occurs)

Figure 13.30 shows an operation example when a one-shot pulse is terminated on underflow. In this example, compare match A as the counter starting trigger is selected and the counter stopping trigger is not selected (TRGSELD00 in DCRD0). Both compare matches A and B as the source of the output signal (OSSD00 in TIOR1D0) are selected. Output levels of 0 for compare match A (IOAD00) and 1 for compare match B (IOBD00) are selected.

Underflow is detected and the underflow flag (UDFD00) in TSRD0 is set on the first edge of the down counter clock after the value in timer down counter D00 (DCNTD00) is H'000000. At the same time, the one-shot pulse output is terminated. The width of the pulse output on pin TOD00B is equal to the value set in DCNTD00 before counting down.

## 13.21 Overview of Timer E

Timer E consists of six subblocks that generate PWM outputs. The subblocks are identical each other. Timer E has the following functions.

- Output of waveform with a duty cycle of 0 to 100% by setting cycle-setting register and duty cycle-setting register
- The values of the cycle-setting and duty-cycle-setting registers are updated every PWM cycle.
   The values in the cycle reload register and duty cycle reload register are reloaded as update data. The reloading function can be enabled and disabled.
- Forcible termination of PWM cycle by writing H'0000 to the counter
- On-state duty (active-high output) and off-state duty (active-low output) modes available
- Interrupt requests can be issued on cycle match (compare match between cycle setting register and timer counter), that is, interrupts are issued every cycle.
- Cycle matches of channel 0 can be used as DMAC activation interrupts, which are automatically cleared by the ACK signal.

#### 13.21.1 Block Diagram of Timer E

Timer E consists of six subblocks. Each subblock includes four channels. Each channel has timer counter E (TCNTE), cycle-setting register E (CYLRE), duty cycle setting register E (DTRE), cycle reload register E (CRLDE), duty cycle reload register E (DRLDE), and controller. Each channel has an output pin for PWM waveforms.

Figure 13.31 is a block diagram of timer E.

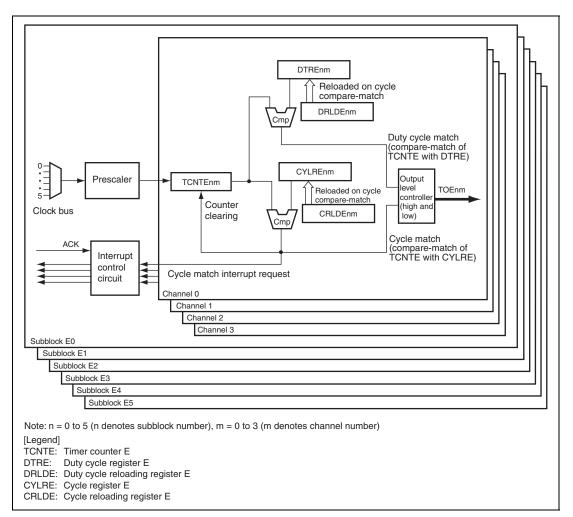


Figure 13.31 Block Diagram of Timer E

#### **Description of Timer E Registers** 13.22

#### **Timer Start Register E (TSTRE)** 13.22.1

TSTRE is an 8-bit readable/writable register that controls subblocks E0 to E5.

The timer E counters run when the timer E enable bit (TEE) in the ATU-III master enable register (ATUENR), timer start register E (TSTRE), and subblock starting register E (SSTRE) must be set.

TSTRE can be read from and written to in byte or word units.

TSTRE is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|---|---|-------|-------|-------|-------|-------|-------|
|                | - | - | STRE5 | STRE4 | STRE3 | STRE2 | STRE1 | STRE0 |
| Initial value: | 0 | 0 | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R | R | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 7, 6 | _        | All 0            | R   | Reserved   |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 5    | STRE5    | 0                | R/W | Subblock E5 Start  |
|      |          |                  |     | 0: Subblock E5 is disabled   |
|      |          |                  |     | 1: Subblock E5 is enabled  |
| 4    | STRE4    | 0                | R/W | Subblock E4 Start  |
|      |          |                  |     | 0: Subblock E4 is disabled   |
|      |          |                  |     | 1: Subblock E4 is enabled  |
| 3    | STRE3    | 0                | R/W | Subblock E3 Start  |
|      |          |                  |     | 0: Subblock E3 is disabled   |
|      |          |                  |     | 1: Subblock E3 is enabled  |
| 2    | STRE2    | 0                | R/W | Subblock E2 Start  |
|      |          |                  |     | 0: Subblock E2 is disabled   |
|      |          |                  |     | 1: Subblock E2 is enabled  |

Apr 01, 2014

|     |          | Initial |     |                            |
|-----|----------|---------|-----|----------------------------|
| Bit | Bit Name | Value   | R/W | Description                |
| 1   | STRE1    | 0       | R/W | Subblock E1 Start          |
|     |          |         |     | 0: Subblock E1 is disabled |
|     |          |         |     | 1: Subblock E1 is enabled  |
| 0   | STRE0    | 0       | R/W | Subblock E0 Start          |
|     |          |         |     | 0: Subblock E0 is disabled |
|     |          |         |     | 1: Subblock E0 is enabled  |

#### 13.22.2 Subblock Starting Registers E0 to E5 (SSTRE0 to SSTRE5)

SSTRE0 to SSTRE5 are 8-bit readable/writable registers that enable and disable the timer counters for four channels of a subblock. Subblocks selected by timer start register E (TSTRE) is enabled. However, both the SSTRE and TEE bits must be set to start counting.

The prescalers run regardless of the counter Enm start bit and are not synchronized with the timing at which TCNTE is started. Therefore, the time from when the counter Enm start bit is set to when TCNTE is incremented for the first time is less than the cycle of the clock of TCNTE.

SSTRE0 to SSTRE5 can be read from and written to in byte or word units.

SSTRE0 to SSTRE5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3           | 2           | 1           | 0           |
|----------------|---|---|---|---|-------------|-------------|-------------|-------------|
|                | - | - | - | - | SSTR<br>En3 | SSTR<br>En2 | SSTR<br>En1 | SSTR<br>En0 |
| 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: n = 0 to 5 (correspond to subblocks E0 to E5)

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 4 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| 3 SSTREn3 0 R/W Counter En3 Start  Enables and disables timer counter En3 (TCNTEn3). When this bit is cleared to 0, TCNTEn3 is disabled. TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn3 is resumed from the retained value.  0: Counter of channel 3 in subblock En is disabled 1: Counter of channel 3 in subblock En is enabled  2 SSTREn2 0 R/W Counter En2 Start  Enables and disables timer counter En2 (TCNTEn2). When this bit is cleared to 0, TCNTEn2 is disabled. TCNTEn2 retains the previous value while it is stopped When this bit is set to 1, TCNTEn2 is resumed from the retained value.  0: Counter of channel 2 in subblock En is disabled  1: Counter of channel 2 in subblock En is enabled  1 SSTREn1 0 R/W Counter En1 Start  Enables and disables timer counter En1 (TCNTEn1). When this bit is cleared to 0, TCNTEn1 is disabled. TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled |     |          | Initial |     |  |
|---|-----|----------|---------|-----|--|
| Enables and disables timer counter En3 (TCNTEn3).  When this bit is cleared to 0, TCNTEn3 is disabled.  TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn3 is resumed from the retained value.  0: Counter of channel 3 in subblock En is disabled  1: Counter of channel 3 in subblock En is enabled  2 SSTREn2 0 R/W Counter En2 Start  Enables and disables timer counter En2 (TCNTEn2).  When this bit is cleared to 0, TCNTEn2 is disabled.  TCNTEn2 retains the previous value while it is stopped When this bit is set to 1, TCNTEn2 is resumed from the retained value.  0: Counter of channel 2 in subblock En is enabled  1 SSTREn1 0 R/W Counter En1 Start  Enables and disables timer counter En1 (TCNTEn1).  When this bit is cleared to 0, TCNTEn1 is disabled.  TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled   | Bit | Bit Name | Value   | R/W | Description  |
| When this bit is cleared to 0, TCNTEn3 is disabled. TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn3 is resumed from the retained value.  0: Counter of channel 3 in subblock En is disabled 1: Counter of channel 3 in subblock En is enabled  2 SSTREn2 0 R/W Counter En2 Start Enables and disables timer counter En2 (TCNTEn2). When this bit is cleared to 0, TCNTEn2 is disabled. TCNTEn2 retains the previous value while it is stopped When this bit is set to 1, TCNTEn2 is resumed from the retained value.  0: Counter of channel 2 in subblock En is disabled 1: Counter of channel 2 in subblock En is enabled  1 SSTREn1 0 R/W Counter En1 Start Enables and disables timer counter En1 (TCNTEn1). When this bit is cleared to 0, TCNTEn1 is disabled. TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled   | 3   | SSTREn3  | 0       | R/W | Counter En3 Start  |
| 1: Counter of channel 3 in subblock En is enabled  2 SSTREn2 0 R/W Counter En2 Start  Enables and disables timer counter En2 (TCNTEn2).  When this bit is cleared to 0, TCNTEn2 is disabled.  TCNTEn2 retains the previous value while it is stopped When this bit is set to 1, TCNTEn2 is resumed from the retained value.  0: Counter of channel 2 in subblock En is disabled  1: Counter of channel 2 in subblock En is enabled  1 SSTREn1 0 R/W Counter En1 Start  Enables and disables timer counter En1 (TCNTEn1).  When this bit is cleared to 0, TCNTEn1 is disabled.  TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled  |     |          |         |     | When this bit is cleared to 0, TCNTEn3 is disabled. TCNTEn3 retains the previous value while it is stopped. When this bit is set to 1, TCNTEn3 is resumed from the |
| 2 SSTREn2 0 R/W Counter En2 Start  Enables and disables timer counter En2 (TCNTEn2). When this bit is cleared to 0, TCNTEn2 is disabled. TCNTEn2 retains the previous value while it is stopped When this bit is set to 1, TCNTEn2 is resumed from the retained value.  0: Counter of channel 2 in subblock En is disabled  1: Counter of channel 2 in subblock En is enabled  1 SSTREn1 0 R/W Counter En1 Start  Enables and disables timer counter En1 (TCNTEn1). When this bit is cleared to 0, TCNTEn1 is disabled. TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled   |     |          |         |     | 0: Counter of channel 3 in subblock En is disabled   |
| Enables and disables timer counter En2 (TCNTEn2). When this bit is cleared to 0, TCNTEn2 is disabled. TCNTEn2 retains the previous value while it is stopped When this bit is set to 1, TCNTEn2 is resumed from the retained value.  0: Counter of channel 2 in subblock En is disabled 1: Counter of channel 2 in subblock En is enabled  1 SSTREn1 0 R/W Counter En1 Start Enables and disables timer counter En1 (TCNTEn1). When this bit is cleared to 0, TCNTEn1 is disabled. TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled  |     |          |         |     | 1: Counter of channel 3 in subblock En is enabled  |
| When this bit is cleared to 0, TCNTEn2 is disabled. TCNTEn2 retains the previous value while it is stopped When this bit is set to 1, TCNTEn2 is resumed from the retained value.  0: Counter of channel 2 in subblock En is disabled 1: Counter of channel 2 in subblock En is enabled  1 SSTREn1 0 R/W Counter En1 Start  Enables and disables timer counter En1 (TCNTEn1). When this bit is cleared to 0, TCNTEn1 is disabled. TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled   | 2   | SSTREn2  | 0       | R/W | Counter En2 Start  |
| 1: Counter of channel 2 in subblock En is enabled  1 SSTREn1 0 R/W Counter En1 Start  Enables and disables timer counter En1 (TCNTEn1).  When this bit is cleared to 0, TCNTEn1 is disabled.  TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled   |     |          |         |     | When this bit is cleared to 0, TCNTEn2 is disabled. TCNTEn2 retains the previous value while it is stopped. When this bit is set to 1, TCNTEn2 is resumed from the |
| 1 SSTREn1 0 R/W Counter En1 Start  Enables and disables timer counter En1 (TCNTEn1).  When this bit is cleared to 0, TCNTEn1 is disabled.  TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled  |     |          |         |     | 0: Counter of channel 2 in subblock En is disabled   |
| Enables and disables timer counter En1 (TCNTEn1). When this bit is cleared to 0, TCNTEn1 is disabled. TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value. 0: Counter of channel 1 in subblock En is disabled  |     |          |         |     | 1: Counter of channel 2 in subblock En is enabled  |
| When this bit is cleared to 0, TCNTEn1 is disabled. TCNTEn3 retains the previous value while it is stopped When this bit is set to 1, TCNTEn1 is resumed from the retained value.  0: Counter of channel 1 in subblock En is disabled   | 1   | SSTREn1  | 0       | R/W | Counter En1 Start  |
|   |     |          |         |     | When this bit is cleared to 0, TCNTEn1 is disabled. TCNTEn3 retains the previous value while it is stopped. When this bit is set to 1, TCNTEn1 is resumed from the |
|   |     |          |         |     | 0: Counter of channel 1 in subblock En is disabled   |
| 1: Counter of channel 1 in subblock En is enabled   |     |          |         |     | 1: Counter of channel 1 in subblock En is enabled  |
| 0 SSTREn0 0 R/W Counter En0 Start   | 0   | SSTREn0  | 0       | R/W | Counter En0 Start  |
|   |     |          |         |     | When this bit is cleared to 0, TCNTEn0 is disabled. TCNTEn0 retains the previous value while it is stopped. When this bit is set to 1, TCNTEn0 is resumed from the |
| 0: Counter of channel 0 in subblock En is disabled  |     |          |         |     | 0: Counter of channel 0 in subblock En is disabled   |
| 1: Counter of channel 0 in subblock En is enabled   |     |          |         |     | 1: Counter of channel 0 in subblock En is enabled  |

Note: n = 0 to 5 (n denotes subblock number)

### 13.22.3 Prescaler Registers E0 to E5 (PSCRE0 to PSCRE5)

PSCRE0 to PSCRE5 are 8-bit readable/writable registers. Each subblock of timer E has one prescaler that divides the frequency of the clock supplied via the clock bus. The register sets the division ratio of the prescalers.

When the value in prescaler register E (PSCRE) is changed, the prescaler updates the value on its underflow. Timer counter E (TCNTE) in the same block is driven by the clock output from prescaler E.

The settable value in prescaler register E (PSCRE) ranges from H'0 to H'7. The division ratio is given below.

Division ratio of prescaler = 
$$\frac{1}{PSCEn[2:0] + 1}$$
 (Settable value: 1/1 to 1/8)

A duty cycle of 50% for the prescaler E output clock is not guaranteed. The high level width is equal to the cycle of the P $\phi$  clock and a low level is output in the remaining cycle of the prescaler E output clock.

Prescaler E runs when the TEE bit in the ATU-III master enable register (ATUENR) and the subblock E start bit (STRE) in timer start register E (TSTRE) are both set to 1.

PSCRE0 to PSCRE5 can be read from and written to in byte or word units.

PSCRE0 to PSCRE5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2   | 1      | 0   |
|----------------|---|---|---|---|---|-----|--------|-----|
|                | - | - | - | - | - | PS  | SCEn[2 | :0] |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0   | 0      | 0   |
| R/W:           | R | R | R | R | R | R/W | R/W    | R/W |

Note: n = 0 to 5 (correspond to subblocks E0 to E5)

|        |          | Initial |     |  |
|--------|----------|---------|-----|--|
| Bit    | Bit Name | Value   | R/W | Description  |
| 7 to 3 | _        | All 0   | R   | Reserved   |
|        |          |         |     | These bits are always read as 0. The write value should always be 0. |
| 2 to 0 | PSCEn    | 000     | R/W | Division Ratio   |
|        | [2:0]    |         |     | These bits store the division ratio of the prescaler.                |

# 13.22.4 Timer Control Register E0 to E5 (TCRE0 to TCRE5)

TCRE0 to TCRE5 are 8-bit readable/writable registers that select the counter clock of prescaler E from clock-bus lines 0 to 5. Timer counter E (TCNTE) is driven by the clock output from prescaler E.

TCRE0 to TCRE5 can be read from and written to in byte or word units.

TCRE0 to TCRE5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2   | 1      | 0     |
|----------------|---|---|---|---|---|-----|--------|-------|
|                | - | - | - | - | 1 | CK  | SELEn[ | [2:0] |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0   | 0      | 0     |
| R/W:           | R | R | R | R | R | R/W | R/W    | R/W   |

Note: n = 0 to 5 (correspond to subblocks E0 to E5)

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7 to 3 | _        | All 0            | R   | Reserved  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.            |
| 2 to 0 | CKSELEn  | 000              | R/W | TCNTEn Clock Select   |
|        | [2:0]    |                  |     | These bits select the counter clock of prescaler E from clock-bus lines 0 to 5. |
|        |          |                  |     | 000: Clock-bus line 0 is selected as counter clock of prescaler E               |
|        |          |                  |     | 001: Clock-bus line 1 is selected as counter clock of prescaler E               |
|        |          |                  |     | 010: Clock-bus line 2 is selected as counter clock of prescaler E               |
|        |          |                  |     | 011: Clock-bus line 3 is selected as counter clock of prescaler E               |
|        |          |                  |     | 100: Clock-bus line 4 is selected as counter clock of prescaler E               |
|        |          |                  |     | 101: Clock-bus line 5 is selected as counter clock of prescaler E               |
|        |          |                  |     | 11x: Reserved   |

[Legend]

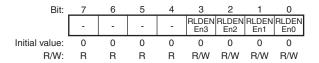
x: Don't care

#### 13.22.5 Reload Control Registers E0 to E5 (RLDCRE0 to RLDCRE5)

RLDCRE0 to RLDCRE5 are 8-bit readable/writable registers that enable and disable the reload function.

RLDCRE0 to RLDCRE5 can be read from and written to in byte or word units.

RLDCRE0 to RLDCRE5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Note: n = 0 to 5 (correspond to subblocks E0 to E5)

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7 to 4 | _        | All 0            | R   | Reserved  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.                                    |
| 3      | RLDENEn3 | 0                | R/W | Reload Enable Enm   |
| 2      | RLDENEn2 | 0                | R/W | These bits enable and disable the function with which the   |
| 1      | RLDENEn1 | 0                | R/W | <ul> <li>duty cycle-setting and cycle-setting registers is reloaded</li> <li>on cycle match.</li> </ul> |
| 0      | RLDENEn0 | 0                | R/W | 0: Reload function on cycle match is disabled   |
|        |          |                  |     | 1: Reload function on cycle match is enabled  |

Note: n = 0 to 5 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

# 13.22.6 Timer Status Registers E0 to E5 (TSRE0 to TSRE5)

TSRE0 to TSRE5 are 8-bit readable/writable registers that indicate occurrence of a cycle match and TCNTE overflow. The cycle match is a compare match between cycle setting register E (CYLRE) and timer counter E (TCNTE).

TSRE0 to TSRE5 can be read from and written to in byte or word units.

TSRE0 to TSRE5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | OVF<br>En3 | OVF<br>En2 | OVF<br>En1 | OVF<br>En0 | CMF<br>En3 | CMF<br>En2 | CMF<br>En1 | CMF<br>En0 |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |

 $R/W \colon R/(W) * R/(W)$ 

Note: n = 0 to 5 (correspond to subblocks E0 to E5)

\* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

|     |          | Initial |        |  |  |  |  |  |  |
|-----|----------|---------|--------|--|--|--|--|--|--|
| Bit | Bit Name | Value   | R/W    | Description  |  |  |  |  |  |
| 7   | OVFEn3   | 0       | R/(W)* | Overflow Flag Enm  |  |  |  |  |  |
| 6   | OVFEn2   | 0       | ` ,    | These bits are set to 1 when timer counter E (TCNTE) overflows. This flag cannot be set to 1 by software.  |  |  |  |  |  |
| 5   | OVFEn1   | 0       | R/(W)* | Overflow account when the account is in a constant of while it   |  |  |  |  |  |
| 4   | OVFEn0   | 0       | R/(W)* | Overflow occurs when the counter is incremented while it is H'FFFF. Writing H'0000 to the counter has no effect on these bits.   |  |  |  |  |  |
|     |          |         |        | When writing a value to the counter and incrementation occur simultaneously while the counter is H'FFFF, the overflow flag is set to 1 but the counter value is changed to the written value instead of H'0000.  |  |  |  |  |  |
|     |          |         |        | No interrupt corresponds to these flags. As to cycle match, since the counter is cleared to H'0001, overflow will not occur. However, it may occur when the value in the cyclesetting register is changed during counter in operation.   |  |  |  |  |  |
|     |          |         |        | When overflow and cycle match occur simultaneously, overflow is not detected (the counter is incremented while it is H'FFFF and CYLREn is H'FFFF). In this case, only the cycle match is handled. If the counter value is H'0001 and the reload function is enabled, cycle reload or duty-cycle reload is performed. |  |  |  |  |  |
|     |          |         |        | 0: Counter E has not overflowed  |  |  |  |  |  |
|     |          |         |        | [Clearing condition]   |  |  |  |  |  |
|     |          |         |        | • When writing 0 to these bits after reading them as 1   |  |  |  |  |  |
|     |          |         |        | 1: Counter E has overflowed  |  |  |  |  |  |
|     |          |         |        | [Setting condition]  |  |  |  |  |  |

When counter E value changes from H'FFFF to H'0000

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |  |  |  |  |  |  |  |  |
|-----|----------|------------------|--------|---|--|--|--|--|--|--|--|--|
| 3   | CMFEn3   | 0                | R/(W)* | Cycle Match Flag Enm  |  |  |  |  |  |  |  |  |
| 2   | CMFEn2   | 0                | R/(W)* | These bits cannot be set to 1 by software. Even if these  |  |  |  |  |  |  |  |  |
| 1   | CMFEn1   | 0                | R/(W)* | bits are 1, meaning that the flag has not been cleared, the next cycle match can be input. In this case, 1 is rewritten to                                    |  |  |  |  |  |  |  |  |
| 0   | CMFEn0   | 0                | R/(W)* | these bits.   |  |  |  |  |  |  |  |  |
|     |          |                  |        | An interrupt request can be issued when the cycle match interrupt enable E bit (CMEEnm) in timer interrupt enable register En (TIEREn) is set to 1.           |  |  |  |  |  |  |  |  |
|     |          |                  |        | To clear these bits, write 0 to these bits after reading them as 1. Flags for channel 0 in each subblock (CMFEn0) is automatically cleared by the ACK signal. |  |  |  |  |  |  |  |  |
|     |          |                  |        | 0: Cycle match has not occurred   |  |  |  |  |  |  |  |  |
|     |          |                  |        | [Clearing conditions]   |  |  |  |  |  |  |  |  |
|     |          |                  |        | • When writing 0 to these bits after reading them as 1  |  |  |  |  |  |  |  |  |
|     |          |                  |        | <ul> <li>When the ACK signal (status clearing) is asserted by<br/>the DMAC (supported only by channel 0)</li> </ul>   |  |  |  |  |  |  |  |  |
|     |          |                  |        | 1: Cycle match has occurred   |  |  |  |  |  |  |  |  |
|     |          |                  |        | [Setting condition]   |  |  |  |  |  |  |  |  |
|     |          |                  |        | <ul> <li>When counter E (TCNTEnm) is incremented while it is<br/>the same value as cycle setting register (CYLREnm)</li> </ul>                                |  |  |  |  |  |  |  |  |

Notes: n = 0 to 5 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

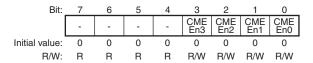
\* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

# 13.22.7 Timer Interrupt Enable Registers E0 to E5 (TIERE0 to TIERE5)

TIERE0 to TIERE5 are 8-bit readable/writable registers that enable and disable interrupt requests occurrence of a cycle match. The cycle match is a compare match between cycle setting register E (CYLRE) and timer counter E (TCNTE).

TIERE0 to TIERE5 can be read from and written to in byte or word units.

TIERE0 to TIERE5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Note: n = 0 to 5 (correspond to subblocks E0 to E5)

| Bit    | Bit Name     | Initial<br>Value | R/W | Description  |
|--------|--------------|------------------|-----|--|
| 7 to 4 | _            | All 0            | R   | Reserved   |
|        |              |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 3 to 0 | CMEEn3       | All 0            | R/W | Cycle Match Interrupt Enable Enm   |
|        | to<br>CMEEn0 |                  |     | These bits enable and disable interrupt requests on cycle match of CYLREnm. When these bits are set to 1, interrupt requests can be issued by using cycle match flag Enm (CMFEnm) in timer status register En (TSREn). |
|        |              |                  |     | 0: CMFEnm interrupt requests are disabled  |
|        |              |                  |     | 1: CMFEnm interrupt requests are enabled   |

Note: n = 0 to 5 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

# 13.22.8 Timer Output Control Registers E0 to E5 (TOCRE0 to TOCRE5)

TOCRE0 to TOCRE5 are 8-bit readable/writable registers that select whether or not a signal on the PWM output pin (TOE) is inverted.

TOCRE0 to TOCRE5 can be read from and written to in byte or word units.

TOCRE0 to TOCRE5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3           | 2           | 1           | 0           |
|----------------|---|---|---|---|-------------|-------------|-------------|-------------|
|                | - | - | - | - | TONE<br>En3 | TONE<br>En2 | TONE<br>En1 | TONE<br>En0 |
| 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: n = 0 to 5 (correspond to subblocks E0 to E5)

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7 to 4 | _        | All 0            | R   | Reserved  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 3      | TONEEn3  | 0                | R/W | TOEnm Output Inversion Select   |
| 2      | TONEEn2  | 0                | R/W | These bits select whether or not a signal on the PWM  |
| 1      | TONEEn1  | 0                | R/W | output pin (TOE) is inverted.   |
| 0      | TONEEn0  | 0                | R/W | The output signal is inverted on the first edge of the P∮ clock after duty modes are switched by the timer output control register. The operating state of the counter (TCNTEnm) has no effect on the mode switching. |
|        |          |                  |     | The initial level on the PWM output pin is low (TONEEnm = 0).   |
|        |          |                  |     | 0: Signal is output as is on PWM output pin (TOEnm)   |
|        |          |                  |     | 1: Inverted signal is output on PWM output pin (TOEnm)  |

Note: n = 0 to 5 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

#### 13.22.9 Timer Counters E00 to E53 (TCNTE00 to TCNTE53)

TCNTE00 to TCNTE53 are 16-bit readable/writable registers that are started by setting the TEE bit in the ATU-III master enable register (ATUENR), the subblock En start bit (STREn) in the timer start register E (TSTRE), and the counter Enm start bit (SSTREnm) in subblock starting register En (SSTREn).

The counter clock is selected by the TCNTEn clock select bits (CKSELEn[2:0]) in timer control register En (TCREn), and prescaler register En (PSCREn) of timer E.

These counters are initialized to H'0001 on cycle match with cycle setting register Enm (CYLREnm). For example, when the value in the cycle setting register is N and the counter value is to be incremented from N to N + 1, the counter value is changed to 1. This enables counting from 1 to N and PWM pulses with the cycle time of N is produced.

These counters can count from H'0001 to H'FFFF (when the cycle setting register value is H'FFFF).

When writing H'0000 to these counters, a PWM cycle is terminated and a new PWM cycle is started in the next clock cycle. While the counter value holds H'0000, the PWM output retains the previous value and outputs a level of 1 at the beginning of the new cycle. When the PWM cycle is terminated before duty cycle match, the duty cycle for that PWM cycle is 100% (1 is always output), that is, a level of 0 will not be output between PWM cycles. For details on writing H'0000 to these counters, see figure 13.33.

When TCNTEnm or CYLREnm is rewritten during the counter in operation, a cycle match may not occur even if the counter value reaches H'FFFF. In this case, the counter value is changed from H'FFFF to H'0000 in the next counter clock cycle. A PWM cycle is terminated in a way similar to writing H'0000. The counter value is incremented to H'0001 and a new PWM cycle is started. When the reload function is enabled, reloading of the cycle or duty cycle is also performed.

TCNTE00 to TCNTE53 can be read from and written to in word units.

TCNTE00 to TCNTE53 are initialized to H'0001 by a power-on reset or a transition to the hardware standby mode.

Apr 01, 2014

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

Notes: 1. n = 0 to 5 (correspond to subblocks E0 to E5) 2. m = 0 to 3 (correspond to channels 0 to 3)

#### 13.22.10 Cycle-Setting Registers E00 to E53 (CYLRE00 to CYLRE53)

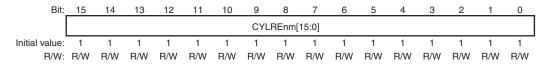
CYLRE00 to CYLRE53 are 16-bit readable/writable registers that store the cycle of PWM. The settable value ranges from H'0001 to H'FFFF.

The value in CYLREnm is constantly compared with the value in the timer counter (TCNTEnm). When they match, the bit in the status register (TSRE) is set to 1 and TCNTEnm is initialized to H'0001. When the RLDENnm bit in the reload control register (RLDCREnm) is set to 1, the values in the cycle reload register (CRLDEnm) and duty cycle reload register (DRLDEnm) are transferred to cycle-setting register (CYLREnm) and duty cycle setting register (DTREnm).

To rewrite to CYLREnm during TCNTEnm in operation, note that the value to be set may lead to malfunction. When TCNTEnm in operation is rewritten, a cycle match may not be detected and TCNTEnm continues to be incremented even if the counter value exceeds the value in CYLREnm. In this case, unwanted PWM waveforms are output.

CYLRE00 to CYLRE53 can be read from and written to in word units.

CYLRE00 to CYLRE53 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.



Notes: 1. n = 0 to 5 (correspond to subblocks E0 to E5) 2. m = 0 to 3 (correspond to channels 0 to 3)

### 13.22.11 Duty Cycle Setting Registers E00 to E53 (DTRE00 to DTRE53)

DTRE00 to DTRE53 are 16-bit readable/writable registers that store the duty cycle of PWM. The settable value ranges from H'0000 to H'FFFF.

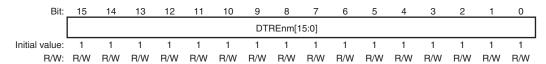
The value in DTREnm is constantly compared with the value in timer counter (TCNTEnm). When they match, the output level on the pin for the corresponding channel becomes low. When the values in CYLREnm and TCNTEnm match while the RLDENnm bit is set to 1, the value in DRLDEnm is reloaded to DTREnm.

The settable value in DTREnm ranges from 0 to the value in CYLREnm. When 0 is set, the duty cycle is 0% and the same value as CYLREnm is set, the duty cycle is 100%. DTREnm must be set to the value less than CYLREnm.

To rewrite to DTREnm during TCNTEnm in operation, note that the value to be set may lead to malfunction. When TCNTEnm in operation is rewritten, a duty cycle match may not be detected. In this case, unwanted PWM waveforms may be output.

DTRE00 to DTRE53 can be read from and written to in word units.

DTRE00 to DTRE53 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.



Notes: 1. n = 0 to 5 (correspond to subblocks E0 to E5)

2. m = 0 to 3 (correspond to channels 0 to 3)

# 13.22.12 Cycle Reload Registers E00 to E53 (CRLDE00 to CRLDE53)

CRLDE00 to CRLDE53 are 16-bit readable/writable register that can be set to H'0001 to H'FFFF as the cycle of PWM outputs.

When the reload function is enabled, the value in this register is transferred to the cycle-setting register (CYLREnm) on cycle match.

CRLDE00 to CRLDE53 can be read from and written to in word units.

CRLDE00 to CRLDE53 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

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

Notes: 1. n = 0 to 5 (correspond to subblocks E0 to E5) 2. m = 0 to 3 (correspond to channels 0 to 3)

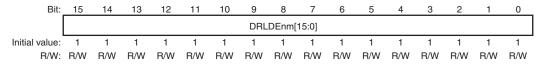
## 13.22.13 Duty Cycle Reload Registers E00 to E53 (DRLDE00 to DRLDE53)

DRLDE00 to DRLDE53 is a 16-bit readable/writable register that can be set to H'0000 to H'FFFF as the duty cycle.

When the reload function is enabled, the value in this register is transferred to the duty cyclesetting register (DTREnm) on cycle match.

DRLDE00 to DRLDE53 can be read from and written to in word units.

DRLDE00 to DRLDE53 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.



Notes: 1. n = 0 to 5 (correspond to subblocks E0 to E5) 2. m = 0 to 3 (correspond to channels 0 to 3)

#### 13.23 **Operations of Timer E**

Timer E consists of timer counter Enm (TCNTEnm), cycle-setting register Enm (CYLREnm), duty cycle setting register Enm (DTREnm), cycle reload register Enm (CRLDEnm), and duty cycle reload register Enm (DRLDEnm). Timer E can be used as a PWM timer.

TCNTEnm starts counting up when a channel is selected by subblock starting register En (SSTREn) after a subblock is selected by timer start register E (TSTRE). A logical zero level is output on pin TOEnm on the first edge of the counter clock after TCNTEnm matches duty cycle setting register Enm (DTREnm), or a logical one level is output on pin TOEnm on the first edge of the counter clock after TCNTEnm matches cycle setting register Enm (CYLREnm). After a match with the cycle setting register, the counter is set to H'0001 on the next counter clock edge and starts counting up again.

Subsequently duty-cycle and cycle match are repeated, producing a PWM output on pin TOEnm.

However, externally output level retains an initial value of 0 for one cycle which is from starting up the counter to the first cycle match.

The settable PWM cycle ranges from H'0001 to H'FFFF. The settable duty cycle ranges from 0% to 100%. When the duty cycle setting register is set to H'0000, the output level is 0 and remains unchanged (duty cycle = 0%). When the values in duty cycle setting register and cycle setting register are the same, the output level is 1 and remains unchanged (duty cycle = 100%). The value in duty cycle setting register must be equal to or less than the value in cycle setting register.

Figure 13.32 shows an operation example of PWM timer outputs for channel 0 in subblock E0. In this example, the duty cycle is changed every PWM cycle in the order of 75%, 67%, 0%, and 100%.

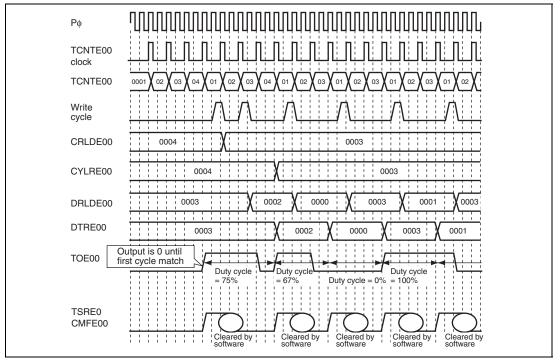


Figure 13.32 Operation of PWM (1)

The duty cycle setting and cycle setting registers have respective reload registers. When the values in the up-counter and cycle-setting register match, the values in the duty cycle reload and cycle reload registers are loaded to the duty cycle setting and cycle setting registers. The loaded data is updated in the next PWM cycle after loading. The reload function is enabled and disabled by the reload enable bit (RLDCREn) in the reload control register (RLDENEnm).

In timer E, a PWM output cycle is terminated by writing H'0000 to the counter (TCNTEnm). The counter value is changed from H'0000 to H'0001 on the next counter clock and the counter is restarted. When the counter value is changed, the values in the duty cycle reload and cycle reload registers are loaded to the duty cycle setting and cycle setting registers.

Figure 13.33 shows the PWM output cycle terminated by writing H'0000 to the counter and the counter restarted. The counter value is cleared to H'0000 by the writing. The output waveform (TOE00) is not changed. When the reload function is enabled after the writing, the values in the duty cycle reload and cycle reload registers are loaded to the duty cycle setting and cycle setting registers. At the same time, counting is restarted and PWM output is also restarted.

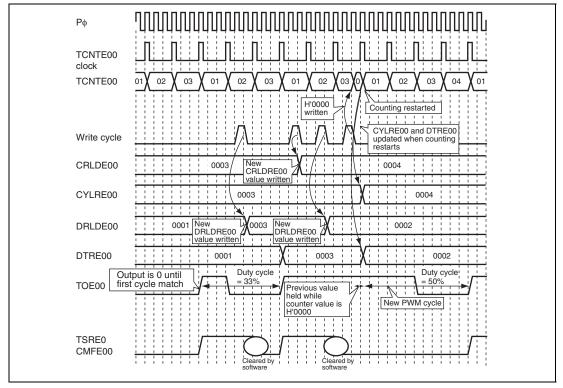


Figure 13.33 Operation of PWM (2)

A waveform is output in off-state duty (active-low output) mode by selecting the off-state duty mode in timer output control register En (TOCREn). The output waveform on pin TOEnm is inverted on the next  $P\phi$  clock cycle after setting.

Figure 13.34 shows an example of a waveform when switching on- and off-state duty modes. By selecting the off-state duty mode before the counter is started, the initial output level on the PWM output pin TOE00 is 1. After the counter started until the first cycle match, the level on pin TOE00 retains 1. On the following cycle match and duty cycle match, the output levels are alternated. When the PWM cycle is forcibly terminated by writing H'0000 to the counter, TOE00 retains the previous value. At the timing in which the counter is incremented to H'0001, a new PWM cycle is started.

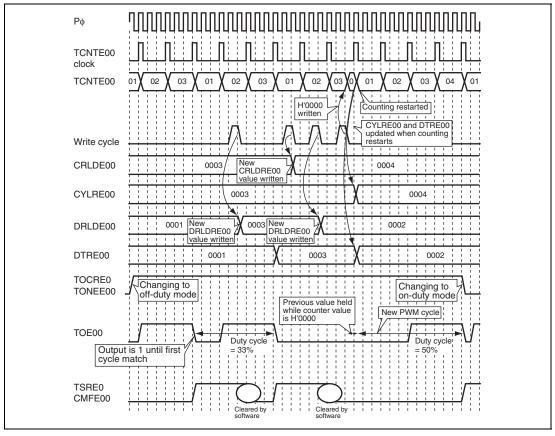


Figure 13.34 Operation of PWM (3)

### 13.24 Overview of Timer F

The timer F consists of 20 subblocks, featuring functions shown below.

- Edge counting in a specified period
   Counts the number of edges input to the external input pin (TIFnA)
- Valid edge interval counting
   Measures time until a specified number of edges is input to the external pin (TIFnA).
- Measurement of time during high/low input levels
   Measures a toal amount of time when a high or low level is input to the external input pin (TIFnA). The duration of measurement is designated as the number of pulses input to the external pin.
- Measurement of PWM input waveform timing
   Measures the off-duty period and cycle time of the PWM waveform input to the external pin (TIFnA). The duration of measurement is designated as the number of PWM cycles input to the external pin.
- Rotation speed/pulse measurement (for the subblock 12 to 15 only)
   Every time an edge is input to the external pin (TIFnA), the following values are retained—
  edge count, time stamp at edge input, edge input interval (cycle), and high/low input level
  immediately before input.
- Up/down event count (for the subblock 0 to 2 only)
   TIFnA of the two external pins (TIFnA, TIFnB) is used to count as the count source. TIFnB switches between upcounting and downcounting.
- Four-time multiplication event count (for the subblock 0 to 2 only)

  Counting operation is executed using two external input pins (TIFnA, TIFnB) as the count sources. Signals in the pins switch between upcounting and downcounting.

Input signals from the external input pins TIFnA and TIFnB can be subject to the noise cancellation function using the input cancellation function.

### 13.24.1 Block Diagram

The timer F consists of 20 subblocks. Each subblock consists of such units as two 24-bit time counters (ECNTAFn, ECNTCFn), three 24-bit general registers (GRAFn, GRCFn, GRDFn), 16-bit event counter (ENCTBFn), 16-bit general register (GRBFn), input processing unit (edge detection, noise canceller), controllers etc.

Figure 13.35 is a block diagram of timer F.

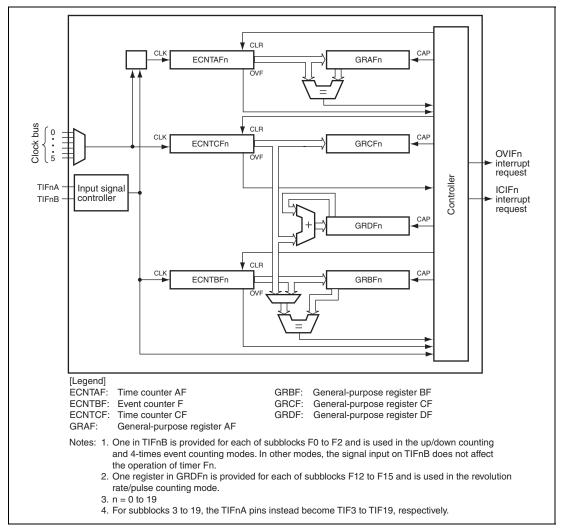


Figure 13.35 Block Diagram of Subblocks of Timer F

### 13.24.2 Interrupts

The timer F can output two types of interrupts totaling 40 interrupts.

## • OVIF0 to OVIF19 interrupts

An interrupt is output when one of the three counters (ECNTAFn, ECNTBFn, ECNTCFn) in the subblock Fn has overflown or underflown (only in ECNTBFn). To which counter the interrupt belongs can be known by referring to the timer status register F (ISRF). This request is received by the INTC module and the designated processing is performed.

## • ICIF0 to ICIF19 interrupts

The interrupt is output when a count value capturing in the subblock Fn occurs. This request is received by the A-DMAC or INTC module. DMA transfer by A-DMAC enables to transfer captured data obtained by using compare match as a trigger to the on-chip SRAM or perform designated processing by interrupts. For details on DMA transfer by A-DMAC, see section 12, Automotive Direct Memory Access Controller (A-DMAC).

# 13.25 Description of Timer F Registers

### 13.25.1 Timer Start Register F (TSTRF)

TSTRF is a 32-bit readable/writable register that specifies whether to operate or stop each subblock (timer F0 to F19) in the timer F. Count operation is not executed unless TFE bit in ATU-III master enable register (ATUENR) is enabled even if the start bit in timer F is set to enable the count operation.

TSTRF can be read from and written to in byte, word, or longword units. However, the execution of access to the register as a longword unit is divided into two operations, i.e. reading or writing the respective words. Accordingly, the bits of both the higher and lower-order words cannot be accessed in the same clock cycle.

TSTRF is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31         | 30         | 29         | 28         | 27         | 26         | 25        | 24        | 23        | 22        | 21        | 20        | 19         | 18         | 17         | 16         |
|----------------|------------|------------|------------|------------|------------|------------|-----------|-----------|-----------|-----------|-----------|-----------|------------|------------|------------|------------|
|                | -          | -          | -          | -          | -          | -          | -         | -         | -         | -         | -         | -         | STRF<br>19 | STRF<br>18 | STRF<br>17 | STRF<br>16 |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0         | 0         | 0         | 0         | 0         | 0         | 0          | 0          | 0          | 0          |
| R/W:           | R          | R          | R          | R          | R          | R          | R         | R         | R         | R         | R         | R         | R/W        | R/W        | R/W        | R/W        |
| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9         | 8         | 7         | 6         | 5         | 4         | 3          | 2          | 1          | 0          |
| DIL.           | 15         | 14         | 13         | 12         |            | 10         |           |           |           |           |           | 4         |            |            |            |            |
|                | STRF<br>15 | STRF<br>14 | STRF<br>13 | STRF<br>12 | STRF<br>11 | STRF<br>10 | STRF<br>9 | STRF<br>8 | STRF<br>7 | STRF<br>6 | STRF<br>5 | STRF<br>4 | STRF<br>3  | STRF<br>2  | STRF<br>1  | STRF<br>0  |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0         | 0         | 0         | 0         | 0         | 0         | 0          | 0          | 0          | 0          |
| R/W:           | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W       | R/W       | R/W       | R/W       | R/W       | R/W       | R/W        | R/W        | R/W        | R/W        |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 31 to 20 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Dir     | Dit Name  | Initial | D AM | Paradation  |
|---------|-----------|---------|------|---|
| Bit     | Bit Name  | Value   | R/W  | Description   |
| 19 to 0 | STRF19 to | All 0   | R/W  | Counter Fn Start  |
|         | STRF0     |         |      | These bits specify whether to operate or stop two time counters in subblocks (ECNTAFn, ECNTCFn) and event counter (ECNTBFn).  Counter value is retained at stop state. When this bit is set to 1 once again, the operation starts at the retained value. Count operation is not executed unless TFE bit in ATU-III master enable register (ATUENR) is enabled even if the start bit in timer F is set enable the count operation.  0: Stop the counting operation of ECNTAFn, ECNTBFn, and ECNTCFn. |
|         |           |         |      |   |
|         |           |         |      | <ol> <li>Enable the counting operation of ECNTAFn, ECNTBFn,<br/>and ECNTCFn.</li> </ol>   |
|         |           |         |      | Note: The prescaler is operating regardless of the setting of the counter F start bit, and not initialized at the start of counter. Therefore, during the time between the activation and the start of actual count operation by the above counter, hardware-related uncertainty shorter than the period of selected count source (resolution) accompanies  |

## 13.25.2 Noise Canceller Control Register F (NCCRF)

NCCRF is a 32-bit readable/writable register that specifies to enable/disable the noise canceller in the subblocks F0 to F19.

NCCRF can be read from and written to in byte, word, or longword units. However, the execution of access to the register as a longword unit is divided into two operations, i.e. reading or writing the respective words. Accordingly, the bits of both the higher and lower-order words cannot be accessed in the same clock cycle.

NCCRF is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31         | 30         | 29         | 28         | 27         | 26         | 25        | 24        | 23        | 22        | 21        | 20        | 19         | 18         | 17         | 16         |
|----------------|------------|------------|------------|------------|------------|------------|-----------|-----------|-----------|-----------|-----------|-----------|------------|------------|------------|------------|
|                | -          | -          | -          | -          | -          | -          | -         | -         | -         | -         | -         | -         | NCEF<br>19 | NCEF<br>18 | NCEF<br>17 | NCEF<br>16 |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0         | 0         | 0         | 0         | 0         | 0         | 0          | 0          | 0          | 0          |
| R/W:           | R          | R          | R          | R          | R          | R          | R         | R         | R         | R         | R         | R         | R/W        | R/W        | R/W        | R/W        |
| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9         | 8         | 7         | 6         | 5         | 4         | 3          | 2          | 1          | 0          |
|                | NCEF<br>15 | NCEF<br>14 | NCEF<br>13 | NCEF<br>12 | NCEF<br>11 | NCEF<br>10 | NCEF<br>9 | NCEF<br>8 | NCEF<br>7 | NCEF<br>6 | NCEF<br>5 | NCEF<br>4 | NCEF<br>3  | NCEF<br>2  | NCEF<br>1  | NCEF<br>0  |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0         | 0         | 0         | 0         | 0         | 0         | 0          | 0          | 0          | 0          |
| R/W:           | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W       | R/W       | R/W       | R/W       | R/W       | R/W       | R/W        | R/W        | R/W        | R/W        |

| Bit      | Bit Name  | Initial<br>Value | R/W   | Description   |
|----------|-----------|------------------|---|---|
| 31 to 20 | _         | All 0            | R   | Reserved These bits are always read as 0. The write value should always be 0.   |
| 19 to 0  | NCEF19 to | All 0            | R/W   | Noise Canceller Enable Fn   |
|          | NCEF0     |                  | Specify to enable/disable the noise canceller in each subblock. Regarding the subblocks F2 to F0, each subblock has noise cancellers TIFnA and TIFnB but enabling/disabling these cancellers cannot be specified independently. Setting the NCEFn bit to 1 enables each noise canceller in TIFnA and TIFnB. |   |
|          |           |                  |   | When the noise canceller function is enabled and the level change in the external input signals (TIFnA, TIFnB) is detected, either subsequent edge cancel mode or preceding edge cancel mode starts according to the setting of noise cancellation mode register (NCMR) in common controller. |

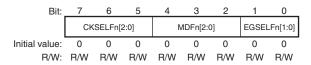
| Bit     | Bit Name           | Initial<br>Value | R/W | Description   |
|---------|--------------------|------------------|-----|---|
| 19 to 0 | NCEF19 to<br>NCEF0 | All 0            | R/W | In subsequent edge cancel mode, when the input signal level change is detected, the change is output as the signal that has passed through noise canceling.  Simultaneously, corresponding noise canceler counters (NCNTFA19 to NCNTFA0 and NCNTFB2 to NCNTFB0) start upcounting. The input signal level change is masked until a compare match occurs between the value in the noise canceler counter and the values in the noise cancel register (NCRFA19 to NCRFA0 and NCRFB2 to NCRFB0). When a compare match occurs, the input signal level at this moment is output as the signal after noise canceling.  |
|         |                    |                  |     | When these bits are cleared to 0 while NCNTFAn and NCNTFBn are in count operation, the count operation continues until a compare match occurs and the level change of the corresponding external input (TIFAn, TIFBn) is kept being masked.   |
|         |                    |                  |     | In preceding edge cancel mode, when the level change of the input signal is detected, the corresponding noise canceler counter (NCNTFA19 to NCNTFA 0 and NCNTFB2 to NCNTFB0) starts upcounting. If a level change of input signal is not detected during the period until a compare match occurs between the value in the noise canceler counter and the values in the noise cancel register (NCRFA19 to NCRFA0 and NCRFB2 to NCRFB0), the level change at the compare match is output as the signal after a noise cancellation. If a noise change is detected, this change is regarded as noise and the noise canceller does not change the signal after a noise cancel regarding that no level change of input signal occurred. |
|         |                    |                  |     | When these bits are cleared to 0 while NCNTFAn and NCNTFBn are in count operation, the count operation continues to keep noise canceling processing until a compare match or input signal change occurs.  |
|         |                    |                  |     | For an operating example in cancel mode, see figures 13.1 and 13.2.   |
|         |                    |                  |     | <ul><li>0: Noise cancel function of TIFnA and TIFnB is disabled.</li><li>1: Noise cancel function of TIFnA and TIFnB is enabled.</li></ul>  |

## 13.25.3 Timer Control Registers F0 to F19 (TCRF0 to TCRF19)

TCRF0 to TCRF19 are 8-bit readable/writable registers that specify the operation mode of the subblocks F0 to F19.

TCRF0 to TCRF19 can be read from and written to in byte units.

TCRF0 to TCRF19 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



| <b></b> | <b>5</b> 0. N | Initial |   | <b></b>          |
|---------|---------------|---------|---|------------------|
| Bit     | Bit Name      | Value   | R/W   | Description      |
| 7 to 5  | CKSELFn       | 000     | R/W   | Clock Select Fn  |
|         | [2:0]         |         | Specify the clock sources for the two time counters (ECNTAFn, ECNTCFn) in the subblocks F0 to F9. Do not specify B'110, B'111. If specified, the operation is not guaranteed. |                  |
|         |               |         |   | 000: Clock bus 0 |
|         |               |         |   | 001: Clock bus 1 |
|         |               |         |   | 010: Clock bus 2 |
|         |               |         |   | 011: Clock bus 3 |
|         |               |         |   | 100: Clock bus 4 |
|         |               |         |   | 101: Clock bus 5 |
|         |               |         |   | 110: Reserved    |
|         |               |         |   | 111: Reserved    |

| Bit    | Bit Name  | Initial<br>Value | R/W | Description  |
|--------|-----------|------------------|-----|--|
| 4 to 2 | MDFn[2:0] | 000              | R/W | Timer Operation Mode Fn  |
|        |           |                  |     | Specify the operation mode for the corresponding subblocks F0 to F19. There are seven modes: up/down event count, four-time multiplication event count, edge counting in a specified period, valid edge interval counting, measurement of time during high/low input levels, measurement of PWM input waveform timing, and rotation speed/pulse measurement. |
|        |           |                  |     | 000: Edge counting in a specified period   |
|        |           |                  |     | 001: Valid edge interval counting  |
|        |           |                  |     | 010: Measurement of time during high/low input levels  |
|        |           |                  |     | 011: Reserved  |
|        |           |                  |     | 100: Measurement of PWM input waveform timing  |
|        |           |                  |     | 101: Rotation speed/pulse measurement  |
|        |           |                  |     | 110: Up/down event count   |
|        |           |                  |     | 111: Four-time multiplication event count  |
|        |           |                  |     | Note: Do not set rotation speed/pulse measurement for subblocks other than 12 to 15.   |

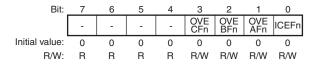
| Bit     | Bit Name      | Initial<br>Value | R/W     | Description  |
|---------|---------------|------------------|---------|--|
| 1, 0    | EGSELFn       | 00               | R/W     | Edge Select Fn   |
|         | [1:0]         |                  |         | Specify the edge sense modes for event input (TIFnA) in the subblocks F0 to F19. Edge detection is done for signals that have passed through the noise canceller. Therefore, edge detection is done to the external input (TIFnA, TIFnB) if the noise cancel function is disabled, and to signals after noise cancel if the noise cancel function is enabled.  |
|         |               |                  |         | While 'measurement of time during high/low input levels' is specified, when this bit selects the falling edge, measurement of time during high level is specified. When this bit selects the rising edge, measurement of time during low level is specified. Do not select both edges.   |
|         |               |                  |         | While 'measurement of PWM input waveform timing' and 'rotation speed/pulse measurement' are specified, when this bit selects the rising edge, the period between the two rising edges is regarded as the PWM cycle and the low-level period is regarded as the off-duty period. If the falling edge is selected, the period between the two falling edges is regarded as the PWM cycle and the high-level period is regarded as the off-duty period. Do not select both edges. |
|         |               |                  |         | When 'up/down event count' mode and 'four-time multiplication event count' mode are specified, be sure to designate both the rising and falling edges. If otherwise selected, the operation is not guaranteed.   |
|         |               |                  |         | 00: Edge detection disabled  |
|         |               |                  |         | 01: Rising edge  |
|         |               |                  |         | 10: Falling edge   |
|         |               |                  |         | 11: Both edges   |
|         |               |                  |         | Note: TIFnB pin is available only when 'up/down event count' and 'four-time multiplication event count' are specified. TIFnB operates always detecting both the rising and falling edges. In other modes, TIFnB does not detect edges.   |
| Note: n | -0 to 19 (cc) | rresnond         | to subh | placks F0 to F19)  |

## 13.25.4 Timer Interrupt Enable Registers F0 to F19 (TIERF0 to TIERF19)

TIERF0 to TIERF19 are 8-bit readable/writable registers that specify whether to enable or disable the interrupt corresponding to the timer status register F (TSRF)

TIERF0 to TIERF19 can be read from and written to in byte units.

TIERF0 to TIERF19 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Note: n = 0 to 19 (correspond to subblocks F0 to F19)

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7 to 4 | _        | All 0            | R   | Reserved These bits are always read as 0. The write value should always be 0.   |
| 3      | OVECFn   | 0                | R/W | Overflow Interrupt Enable CFn*  |
|        |          |                  |     | Specifies whether to enable or disable the interrupt by OVFCFn to the status corresponding to the overflow of the time counter CFn (ECNTCFn) (in 'measurement of PWM input waveform timing' mode) or a compare match between ECNTCFn and GRBFn (in 'rotation speed/pulse measurement' mode) |
|        |          |                  |     | 0: Interrupt by OVFCFn disabled   |
|        |          |                  |     | 1: Interrupt by OVFCFn enabled  |
| 2      | OVEBFn   | 0                | R/W | Overflow Interrupt Enable BFn*  |
|        |          |                  |     | Specifies whether to enable or disable the interrupt by OVBCFn to the status corresponding to the overflow/underflow of the event counter BFn (ECNTBFn).  |
|        |          |                  |     | 0: Interrupt by OVFBFn disabled   |
|        |          |                  |     | 1: Interrupt by OVFBFn enabled  |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 1   | OVEAFn   | 0                | R/W | Overflow Interrupt Enable AFn*  |
|     |          |                  |     | Specifies whether to enable or disable the interrupt by OVFAFn to the status corresponding to the overflow of the time counter AFn (ECNTAFn). |
|     |          |                  |     | 0: Interrupt by OVFAFn disabled   |
|     |          |                  |     | 1: Interrupt by OVFAFn enabled  |
| 0   | ICEFn    | 0                | R/W | Input Capture Interrupt Enable Fn   |
|     |          |                  |     | Specifies whether to enable or disable the interrupt by OVFAFn to the status corresponding to the input capture detection in the subblock Fn. |
|     |          |                  |     | 0: Interrupt by ICFFn disabled  |
|     |          |                  |     | 1: Interrupt by ICFFn enabled   |

Note: \* The overflow of interrupt of the subblock Fn is requested as the logical sum of the interrupts OVFAFn, OVFBFn, and OVFCFn, By referring to TSRFn, which counter generated the interrupt by overflow or underflow can be known.

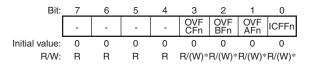
## 13.25.5 Timer Status Registers F0 to F19 (TSRF0 to TSRF19)

TSRF0 to TSRF19 are 8-bit readable/writable registers that indicate overflows in the time counters A and C, overflow or underflow in the event counter, and input capture occurrence.

These flags are interrupt sources and requests the CPU interrupts if the corresponding bits to the timer interrupt enable register F0 to F19 (TIERF0 to TIERF19) enable interrupts.

TSRF0 to TSRF19 can be read from and written to in byte units.

TSRF0 to TSRF19 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



<sup>\*</sup> Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

| Bit    | Bit Name | Initial<br>Value | R/W    | Description  |
|--------|----------|------------------|--------|--|
| 7 to 4 | _        | All 0            | R      | Reserved These bits are always read as 0. The write value should always be 0.  |
| 3      | OVFCFn   | 0                | R/(W)* | Overflow/Compare Match Flag CFn  |
|        |          |                  |        | The values in this flag show different states depending on the operation mode. In 'measurement of PWM input waveform timing', the flag shows the overflow of the time counter C (ECNTCFn). In 'rotation speed/pulse measurement' mode, the flag shows a compare match between ECNTCFn and GRBFn. |
|        |          |                  |        | This flag cannot be set to 1 by software.  |
|        |          |                  |        | 0: No overflow in ECNTCFn  |
|        |          |                  |        | [Clearing condition]   |
|        |          |                  |        | • After reading OVFCFn = 1, 0 is written to OVFCFn.  |
|        |          |                  |        | 1: ECNTCFn overflows   |
|        |          |                  |        | [Setting conditions]   |
|        |          |                  |        | Measurement of PWM input waveform timing mode  |
|        |          |                  |        | When ECNTCFn overflows (H'FFFF FF $\rightarrow$ H'0000 00)   |
|        |          |                  |        | Rotation speed/pulse measurement mode  |
|        |          |                  |        | When values in ECNTCFn and GRBFn (with the value zero extended to lower 8 bits) match  |
| 2      | OVFBFn   | 0                | R/(W)* | Overflow Flag BFn  |
|        |          |                  |        | By this bit, overflow or underflow of the event counter BFn (ECNTBFn) can be monitored. This flag cannot be set to 1 by software.  |
|        |          |                  |        | 0: No overflow or underflow in ECNTBFn   |
|        |          |                  |        | [Clearing condition]   |
|        |          |                  |        | • After reading OVFBFn = 1, 0 is written to OVFBFn.  |
|        |          |                  |        | 1: ECNTBFn overflows or underflows   |
|        |          |                  |        | [Setting condition]  |
|        |          |                  |        | • When ECNTBFn overflows (H'FFFF $\rightarrow$ H'0000) or underflows (H'0000 $\rightarrow$ H'FFFF)   |

| Bit | Bit Name | Initial<br>Value | R/W    | Description  |
|-----|----------|------------------|--------|--|
| 1   | OVFAFn   | 0                |        | Overflow Flag AFn  |
|     |          |                  | ( ,    | By this bit, overflow of the time counter AFn (ECNTAFn) can be monitored. This flag cannot be set to 1 by software.              |
|     |          |                  |        | 0: No overflow in ECNTAFn  |
|     |          |                  |        | [Clearing condition]   |
|     |          |                  |        | • After reading OVFAFn = 1, 0 is written to OVFAFn.  |
|     |          |                  |        | 1: ECNTAFn overflows   |
|     |          |                  |        | [Setting condition]  |
|     |          |                  |        | • When ECNTAFn overflows (H'FFFF FF $\rightarrow$ H'0000 00)   |
| 0   | ICFFn    | 0                | R/(W)* | Input capture Flag Fn  |
|     |          |                  |        | By this bit, the detection state of input capture in the subblock Fn can be monitored. This flag cannot be set to 1 by software. |
|     |          |                  |        | 0: Input capture is not detected in the subblock Fn.   |
|     |          |                  |        | [Clearing conditions]  |
|     |          |                  |        | • After reading ICFFn = 1, 0 is written to ICFFn.  |
|     |          |                  |        | <ul> <li>When the capture output register (CDRF0 to CDRF19)<br/>are read by A-DMAC access.</li> </ul>                            |
|     |          |                  |        | 1: Input capture in subblock Fn detected   |
|     |          |                  |        | [Setting condition]  |
|     |          |                  |        | When input capture is detected in the subblock Fn.    Solve F0 to F10  |

<sup>\*</sup> To clear the flag, only writing 0 after reading 1 is possible. Writing 1 is invalid.

## Timer Counters AF0 to AF19 (ECNTAF0 to ECNTAF19)

ECNTAF0 to ECNTAF19 are 32-bit readable/writable registers.

This register, with one provided to each subblock, executes upcount operation using the input clock. One clock bus from clock buses 0 to 5 can be selected as the input clock according to the setting of the corresponding control register. The input clocks for ECNTAFn and ECNTCFn are the same. Clock source cannot be set independently.

When clearing the counter is done at the countup timing, ECNTAFn is cleared to H'00000100, and to H'00000000 in other cases.

ECNTAF0 to ECNTAF19 can be read from and written to in longword units.

ECNTAF0 to ECNTAF19 are initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30            | 29  | 28    | 27      | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|---------------|-----|-------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     | ECNTAFn[23:8] |     |       |         |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 0   | 0             | 0   | 0     | 0       | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W           | R/W | R/W   | R/W     | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit:           | 15  | 14            | 13  | 12    | 11      | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |               |     | ECNTA | \Fn[7:0 | ]   |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0             | 0   | 0     | 0       | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W           | R/W | R/W   | R/W     | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

|         |          | Initial |     |  |  |  |  |  |  |  |
|---------|----------|---------|-----|--|--|--|--|--|--|--|
| Bit     | Bit Name | Value   | R/W | Description  |  |  |  |  |  |  |
| 31 to 8 | ECNTAFn  | All 0   | R/W | Time Count AFn   |  |  |  |  |  |  |
|         | [23:0]   |         |     | Upcounter A  |  |  |  |  |  |  |
| 7 to 0  | _        | All 0   | R   | Reserved   |  |  |  |  |  |  |
|         |          |         |     | These bits are always read as 0. The write value should always be 0. |  |  |  |  |  |  |

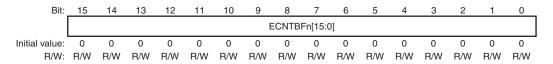
## 13.25.7 Event Counters BF0 to BF19 (ECNTBF0 to ECNTBF19)

ECNTBF0 to ECNTBF19 are 16-bit readable/writable registers. This register, with one provided to each subblock, executes upcount/downcount operation using the input clock. The input clock is given two external input pins (TIFnA, TIFnB). The external pin and edge used to count differs according to the setting of the corresponding control register (operation mode and edge select). The input clock in each mode is listed in table 13.15.

When clearing the counter is done at the count-up timing, ECNTBFn is cleared to H'0001, and to H'0000 in other cases.

ECNTBF0 to ECNTBF19 can be read from and written to in word units.

ECNTBF0 to ECNTBF19 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.



| Bit     | Bit Name | Initial<br>Value | R/W | Description           |
|---------|----------|------------------|-----|-----------------------|
| 15 to 0 | ECNTBFn  | All 0            | R/W | Event Count Fn        |
|         | [15:0]   |                  |     | Up/down event counter |

Table 13.15 Event Counter Input Clock and Count Edge for Each Timer F Operation Mode

| Operation Mode                                   | Input Clock  | Count Edge                                    |
|--|--|---|
| Edge counting in a specified period              | TIFnA  | Selectable by EGSELFn                         |
| Valid edge interval counting                     | TIFnA  | Selectable by EGSELFn                         |
| Measurement of time during high/low input levels | TIFnA  | Selectable by EGSELFn (other than both edges) |
| Measurement of PWM input waveform timing         | TIFnA  | Selectable by EGSELFn (other than both edges) |
| Rotation speed/pulse measurement                 | TIFnA  | Selectable by EGSELFn (other than both edges) |
| Up/down event count                              | TIFnA (Count<br>direction is<br>specified by TIFnB<br>level) | Both rising/falling edges                     |
| Four-time multiplication event count             | TIFnA, TIFnB   | Both rising/falling edges                     |

## 13.25.8 Time Counters CF0 to CF19 (ECNTCF0 to ECNTCF19)

ECNTCF0 to ECNTCF19 are 32-bit readable/writable registers. This register, with one provided to each subblock, is enabled only in 'measurement of PWM input waveform timing' and 'rotation speed/pulse measurement' modes. This register does not execute count operation in other modes. This register executes upcount operation using the input clock. One clock bus from clock buses 0 to 5 can be selected as the input clock according to the setting of the corresponding control register. The input clocks for ECNTAFn and ECNTCFn are the same. Clock source cannot be set independently.

When clearing the counter is done at the external input timing or triggered by ECNTBFn, ECNTCFn is cleared synchronized with ECNTCFn count clock. At this moment, ECNTCFn is cleared to H'0000100.

ECNTCF0 to ECNTCF19 can be read from and written to in longword units.

ECNTAF0 to ECNTAF19 are initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30            | 29  | 28   | 27      | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|---------------|-----|------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     | ECNTCFn[23:8] |     |      |         |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 0   | 0             | 0   | 0    | 0       | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W           | R/W | R/W  | R/W     | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| ъ.,            |     |               |     |      |         |     | _   | _   | _   | _   | _   |     | _   | _   |     |     |
| Bit:           | 15  | 14            | 13  | 12   | 11      | 10  | 9   | 8   |     | 6   | 5   | 4   | 3   | 2   | _1_ | 0   |
|                |     |               |     | ECNT | CFn[7:0 | )]  |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 0   | 0             | 0   | 0    | 0       | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W           | R/W | R/W  | R/W     | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

Note: n = 0 to 19 (correspond to subblocks F0 to F19)

. ... .

|         |          | Initial |     |  |
|---------|----------|---------|-----|--|
| Bit     | Bit Name | Value   | R/W | Description  |
| 31 to 8 | ECNTCFn  | All 0   | R/W | Time Count CFn   |
|         | [23:0]   |         |     | Upcounter C  |
| 7 to 0  | _        | All 0   | R   | Reserved   |
|         |          |         |     | These bits are always read as 0. The write value should always be 0. |

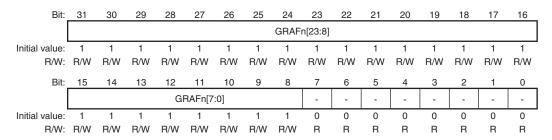
## 13.25.9 General Registers AF0 to AF19 (GRAF0 to GRAF19)

GRAF0 to GRAF19 are 32-bit readable/writable registers. This register, with one provided to each subblock, has two functions such as input capture register and output compare register for the time counter Afn (ECNTAFn).

Do not set GRAFn to H'00000000 to function this register as the compare match register. Note that if H'00000000 is set, incorrect measurement may occur.

GRAF0 to GRAF19 can be read from and written to in longword units.

GRAF0 to GRAF19 are initialized to H'FFFF FF00 by a power-on reset or a transition to the hardware standby mode.



|         |          | Initial |     |   |
|---------|----------|---------|-----|---|
| Bit     | Bit Name | Value   | R/W | Description   |
| 31 to 8 | GRAFn    | All 1   | R/W | General Registers AFn   |
|         | [23:0]   |         |     | Input capture value or output compare match value for the time counter A. |
| 7 to 0  | _        | All 0   | R   | Reserved  |
|         |          |         |     | These bits are always read as 0. The write value should always be 0.      |

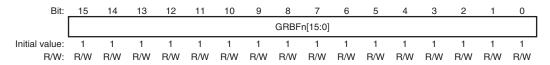
### 13.25.10 General Registers BF0 to BF19 (GRBF0 to GRBF19)

GRBF0 to GRAB19 are 16-bit readable/writable registers. This register, with one provided to each subblock, has two functions such as input capture register and output compare register for the event counter (ECNTBFn).

Do not set GRBFn to H'0000 to function this register as the compare match register. Note that if H'0000 is set, incorrect measurement may occur.

GRBF0 to GRBF19 can be read from and written to in word units.

GRBF0 to GRBF19 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.



| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
|     | GRBFn    | All 1            | R/W | General Registers BFn  |
|     | [15:0]   |                  |     | Input capture value or output compare match value for event counter. |

## 13.25.11 General Registers CF0 to CF19 (GRCF0 to GRCF19)

GRCF0 to GRCB19 are 32-bit readable/writable registers. This register, with one provided to each subblock, has a function as the input capture register for the time counter (ECNTCFn). Triggered by a compare match between ECNTBFn and GRBn (in 'measurement of PWM input waveform timing' mode) or edge input of the TIFnA pin (in 'rotation speed/pulse measurement' mode), ECNTCFn count number is taken in at the next ECNTCFn upcount timing. These registers are valid only in 'measurement of PWM input waveform timing' or 'rotation speed/pulse measurement' mode. Capture operation is not executed in other modes.

GRCF0 to GRCB19 can be read from and written to in longword units.

GRCF0 to GRCB19 are initialized to H'FFFFFF00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30          | 29  | 28  | 27     | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-------------|-----|-----|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     | GRCFn[23:8] |     |     |        |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 1   | 1           | 1   | 1   | 1      | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W:           | R/W | R/W         | R/W | R/W | R/W    | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit:           | 15  | 14          | 13  | 12  | 11     | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |             |     | GI  | RCFn[7 | :0] |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 1   | 1           | 1   | 1   | 1      | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W         | R/W | R/W | R/W    | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 31 to 8 | GRCFn    | All 1            | R/W | General Registers CFn  |
|         | [23:0]   |                  |     | Input capture value for the time counter C                           |
| 7 to 0  | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |

### 13.25.12 General Registers DF12 to DF15 (GRDF12 to GRDF15)

GRDF12 to GRDF15 are 32-bit readable/writable registers. This register is provided to subblocks F12 to F15. Triggered by edge input of the TIFn pin, accumulated number in the time counter CFn (ECNTCFn) is taken in at the next ECNTAFn upcount timing. This register is valid only in 'rotation speed/pulse measurement' mode. Capture operation is not executed in other modes.

GRDF12 to GRDF15 can be read from and written to in longword units.

GRDF12 to GRDF15 are initialized to H'FFFFFF00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31  | 30          | 29  | 28  | 27     | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-------------|-----|-----|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     | GRDFn[23:8] |     |     |        |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 1   | 1           | 1   | 1   | 1      | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| R/W:           | R/W | R/W         | R/W | R/W | R/W    | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit:           | 15  | 14          | 13  | 12  | 11     | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |             |     | GI  | RDFn[7 | :0] |     |     | -   | -   | -   | -   | -   | -   | -   | -   |
| Initial value: | 1   | 1           | 1   | 1   | 1      | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W | R/W         | R/W | R/W | R/W    | R/W | R/W | R/W | R   | R   | R   | R   | R   | R   | R   | R   |

|         |          | Initial |     |  |
|---------|----------|---------|-----|--|
| Bit     | Bit Name | Value   | R/W | Description  |
| 31 to 8 | GRDFn    | All 1   | R/W | General Registers DFn  |
|         | [23:0]   |         |     | Input capture value for the time counter A                           |
| 7 to 0  | _        | All 0   | R   | Reserved   |
|         |          |         |     | These bits are always read as 0. The write value should always be 0. |

### 13.25.13 Capture Output Registers F0 to F19 (CDRF0 to CDRF19)

GRCF0 to GRCB19 are 32-bit readable/writable registers. This register is provided to each subblock. When this register is read, values in GRAFn, GRBFn, or ECNTBFN is read according to the operation mode. A 16-bit value in GRBFn is read from the upper 16 bits in CDRFn. In this case, the lower eight bits in CDRFn are read as 0.

CDRF0 to CDRF19 can be read from and written to in longword units.

CDRF0 to CDRF19 are initialized to H'FFFF0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31         | 30          | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|                |            | CDRFn[23:8] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Initial value: | 1          | 1           | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |
| R/W:           | R          | R           | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |
| Bit:           | 15         | 14          | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|                | CDRFn[7:0] |             |    |    |    |    |    |    | -  | -  | -  | -  | -  | -  | -  | -  |
| Initial value: | 0          | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| R/W:           | R          | R           | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  | R  |

| D.,     | Bit    | Initial  | D 444 | Post today  |
|---------|--------|----------|-------|---|
| Bit     | Name   | Value    | R/W   | Description   |
| 31 to 8 | CDRFn  | H'FFFF00 | R     | Capture Output Register Fn  |
| [23:    | [23:0] |          |       | Data stored in GRAFn or GRBFn is read according to the operation mode. Registers corresponding to various modes are listed below. Writing to these registers are ignored. |
|         |        |          |       | Edge counting in a specified period mode: GRBFn   |
|         |        |          |       | Valid edge interval counting mode: GRAFn  |
|         |        |          |       | Measurement of time during high input levels mode: GRAFn  |
|         |        |          |       | Measurement of time during low input levels mode: GRAFn   |
|         |        |          |       | Measurement of PWM input waveform timing mode: GRAFn  |
|         |        |          |       | Rotation speed/pulse measurement mode: ECNTBFn  |
|         |        |          |       | Up/down event count mode mode: GRBFn  |
|         |        |          |       | Four-time multiplication event count mode: GRBFn  |

| Bit    | Bit<br>Name | Initial<br>Value | R/W | Description  |
|--------|-------------|------------------|-----|--|
| 7 to 0 | _           | All 0            | R   | Reserved   |
|        |             |                  |     | These bits are always read as 0. The write value should always be 0. |

#### 13.25.14 Noise Canceler Counters FA0 to FA19 (NCNTFA0 to NCNTFA19)

NCNTFA0 to NCNTFA19 are 8-bit readable/writable registers. When the noise canceler control register Fn (NCCRFn) enables the noise canceler function, these registers start upcount operation using the count clock for noise canceler supplied by the pre-scaler, triggered by the level change in the external input pin (TIFnA).

The timer F can output two types of interrupts totaling 40 interrupts.

Two types of operation modes — subsequent edge cancel mode and preceding edge cancel mode — can be set according to the setting of the timer F niose cancel mode bit (NCMF) in the noise cancelltion mode register (NCMR) in the common controller.

## • Subsequent edge cancel mode

NCNTFAn starts upcount operation triggered by the level change of input signal in TIFnA under the condition that the NCEFn bit is set to 1 and NCNTFAn is not in count operation. When the count number matches the value in the noise cancel register FAn (NCRFAn), this register stops the count operation, clearing the count value to H'00 synchronizing with the next  $P\phi$  clock. NCNTFAn executes the count operation regardless of the setting of the TFE bit in the ATU-III master enable register (ATUENR).

A level change at the start of count operation is output as it is as the signal that passed through the noise canceling operation. Although this signal is the subject of edge detection, the signal does not change because any input level change is masked until the count number matches the value of NCRFAn.

Even if the NCEFn bit is cleared during the count operation, the count operation continues until the count number matches the value of NCRFAn. The input signal is masked during all that time.

#### Preceding edge cancel mode

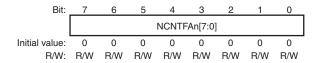
NCNTFAn starts upcount operation triggered by the level change of input signal in TIFnA under the condition that the NCEFn bit is set to 1 and NCNTFAn is not in count operation. When the level change of the input signal occurs during the count operation or the count number matches the value in the noise cancel register FAn (NCRFAn), this register stops the count operation, clearing the count value to H'00 synchronizing with the next P $\phi$  clock. NCNTFAn executes the count operation regardless of the setting of the TFE bit in the ATU-III master enable register (ATUENR).

Signals that passed through the noise canceling operation can change only when the count number matches the value of NCRFAn according to the level change at the count strart. If the count operation stops before the count number matches the value of NCRFAn, signals that passed through the noise canceling operation do not change because the level changes at the start or stop of counting are masked.

Even if the NCEFn bit is cleared during the count operation, the count operation and noise canceling processing continue until a compare match occurs or the level of the input signal changes.

NCNTFA0 to NCNTFA19 can be read from and written to in byte units.

NCNTFA0 to NCNTFA19 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



| Bit    | Bit Name | Initial<br>Value | R/W               | Description            |
|--------|----------|------------------|-------------------|------------------------|
| 7 to 0 | NCNTFAn  | All 0            | R/W               | Noise Cancel Count FAn |
|        | [7:0]    |                  | 8-bit count value |                        |

#### 13.25.15 Noise Canceler Counters FB0 to FB2 (NCNTFB0 to NCNTFB2)

NCNTFB0 to NCNTFB2 are 8-bit readable/writable registers.

These registers are available only in 'up/down event count' mode and '4-time multiplication event count' mode.

When the noise canceler control register Fn (NCCRFn) enables the noise canceler function, these registers start upcount operation using the count clock for noise canceler supplied by the prescaler, triggered by the level change in the external input pin (TIFnB).

Two types of operation modes — subsequent edge cancel mode and preceding rdge cancel mode — can be set according to the setting of the timer F niose cancel mode bit (NCMF) in the noise cancelltion mode register (NCMR) in the common controller.

#### Subsequent edge cancel mode

NCNTFBn starts upcount operation triggered by the level change of input signal in TIFnB under the condition that the NCEFn bit is set to 1 and NCNTFBn is not in count operation. When the count number matches the value in the noise cancel register FBn (NCRFBn), this register stops the count operation, clearing the count value to H'00 synchronizing with the next P clock. NCNTFBn executes the count operation regardless of the setting of the TFE bit in the ATU-III master enable register (ATUENR).

A level change at the start of count operation is output as it is as the signal that passed through the noise canceling operation. Although this signal is the subject of edge detection, the signal does not change because any input level change is masked until the count number matches the value of NCRFBn.

Even if the NCEFn bit is cleared during the count operation, the count operation continues until the count number matches the value of NCRFBn. The input signal is masked during all that time.

## • Preceding edge cancel mode

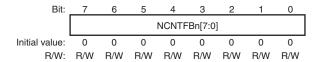
NCNTFBn starts upcount operation triggered by the level change of input signal in TIFnB under the condition that the NCEFn bit is set to 1 and NCNTFBn is not in count operation. When the level change of the input signal occurs during the count operation or the count number matches the value in the noise cancel register FBn (NCRFBn), this register stops the count operation, clearing the count value to H'00 synchronizing with the next P $\phi$  clock. NCNTFBn executes the count operation regardless of the setting of the TFE bit in the ATU-III master enable register (ATUENR).

Signals that passed through the noise canceling operation can change only when the count number matches the value of NCRFBn, according to the level change at the count strart. If the count operation stops before the count number matches the value of NCRFBn, signals that passed through the noise canceling operation do not change because the level changes at the start or stop of counting are masked.

Even if the NCEFn bit is cleared during the count operation, the count operation and noise canceling processing continue until a compare match occurs or the level of the input signal changes.

NCNTFB0 to NCNTFB19 can be read from and written to in byte units.

NCNTFB0 to NCNTFB19 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



| Bit    | Bit Name                | Initial<br>Value | R/W | Description            |
|--------|-------------------------|------------------|-----|------------------------|
| 7 to 0 | 7 to 0 NCNTFBn<br>[7:0] | All 0            | R/W | Noise Cancel Count FBn |
|        |                         |                  |     | 8-bit count value      |

## 13.25.16 Noise Cancel Registers FA0 to FA19 (NCRFA0 to NCRFA19)

NCRFA0 to NCRFA19 are 8-bit readable/writable registers that set the upper limit of the noise canceler counter (NCNTFAn). Noise in the period up to 1.64 ms (when  $P\phi = 20MHz$ ) can be cancelled by setting the register to H'FF.

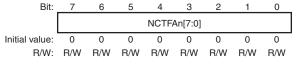
Two types of operation modes — subsequent edge cancel mode and preceding rdge cancel mode — can be set according to the setting of the timer F niose cancel mode bit (NCMF) in the noise cancelltion mode register (NCMR) in the common controller.

- Subsequent edge cancel mode
  - While NCNTFAn is in count operation, the level change of the subsequent input signal is masked. Values in NCNTFAn and NCRFAn are always compared. If a compare match occurs, these registers clear the value in NCNTFAn synchronizing with the next  $P\phi$  clock, stop the count operation, and cancel the masking of the input signal.
- Preceding edge cancel mode

While NCNTFAn is in count operation, noise canceler processing waiting state is entered. Values in NCNTFAn and NCRFAn are always compared. If a compare match occurs, these registers clear the value in NCNTFAn synchronizing with the next  $P\phi$  clock, stop the count operation, and then cancel the masking of the input signal and the noise canceler outputs the input signal that has passed through the noise canceling processing.

NCRFA0 to NCRFA19 can be read from and written to in byte units.

NCRFA0 to NCRFA19 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



| Bit    | Bit Name | Initial<br>Value | R/W | Description                                     |
|--------|----------|------------------|-----|---|
| 7 to 0 | NCTFAn   | All 0            | R/W | Noise Cancel Time FAn                           |
|        | [7:0]    |                  |     | TIFnA noise cancel period (8-bit compare value) |

#### 13.25.17 Noise Cancel Registers FB0 to FB2 (NCRFB0 to NCRFB2)

NCRFB0 to NCRFB2 are 8-bit readable/writable registers that set the upper limit of the noise canceler counter (NCNTFBn). Noise in the period up to 1.64 ms (when  $P\phi = 20MHz$ ) can be cancelled by setting the register to H'FF.

These registers are available only in 'up/down event count' mode and '4-time multiplication event count' mode.

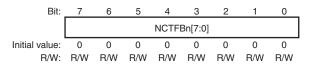
Two types of operation modes — subsequent edge cancel mode and preceding rdge cancel mode — can be set according to the setting of the timer F niose cancel mode bit (NCMF) in the noise cancelltion mode register (NCMR) in the common controller.

- Subsequent edge cancel mode
  - While NCNTFBn is in count operation, the level change of the subsequent input signal is masked. Values in NCNTFBn and NCRFBn are always compared. If a compare match occurs, these registers clear the value in NCNTFBn synchronizing with the next  $P\phi$  clock, stop the count operation, and cancel the masking of the input signal.
- Preceding edge cancel mode

While NCNTFBn is in count operation, noise canceler processing waiting state is entered. Values in NCNTFBn and NCRFBn are always compared. If a compare match occurs, these registers clear the value in NCNTFBn synchronizing with the next Pφ clock, stop the count operation. Simultaneously the noise canceler outputs the input signal that has passed through noise canceling processing.

NCRFB0 to NCRFB2 can be read from and written to in byte units.

NCRFB0 to NCRFB2 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



| Bit    | Bit Name | Initial<br>Value | R/W | Description                                     |
|--------|----------|------------------|-----|---|
| 7 to 0 | NCTFBn   | All 0            | R/W | Noise Cancel Time FBn                           |
|        | [7:0]    |                  |     | TIFnB noise cancel period (8-bit compare value) |

# 13.26 Operations of Timer F

### 13.26.1 Edge Counting

When a period over which edges are counted is set in GRAFn, the number of edges within the period is obtained in GRBFn. When no edge is detected within the period, 0 is set to GRBFn. The period set to count is eqivqlent to the cycle of the ECNTAFn clock (GRAFn value). Operation of the timer Fn is described below. Figure 13.36 shows an operation example. In this example, eight edges are input to 12 cycles of the count source clock. Timer counter ECNTAFn and event counter ECNTBFn are driven by the ECNTAFn and ECNTBFn clocks, respectively.

Operation of registers in edge counting mode is shown below.

- ECNTAFn: Measures time using one of the clock buses 0 to 5. When a compare match is detected, the count value is cleared synchronized with the next Pφ clock.
- ECNTBFn: Counts edges of the signals provided from TIFnA input. Edge types subject to count can be selected from among rising, falling, or both edges. The example given here counts the falling edges. A delay of two cycles in TIFnA occurs because of synchronization processing. When a compare match in ECNTAFn is detected, the count number is cleared synchronized with the next Pφ clock. In a case where edges subject to count are given simultaneously at a count clearing by a compare match, both operations are regarded to be done in one cycle, setting the count value to H'0001. Figure 13.38 shows an example of this.
- GRAFn: Functions as the compare match register for ECNTAFn. A compare match is detected when the count values in ECNTAFn and GRAFn agree.
- GRBFn: Functions as the capture register for ECNTBFn. When a compare match in ECNTAFn is detected, this register captures the ECNTBFn count number synchronizing with the next Pφ clock.
- ICFFn flag: After detecting a compare match in ECNTAFn, sets the ICFFn flag synchronized with the next Pφ clock.
- ECNTCFn, GRCFn, GRDFn: Do not function

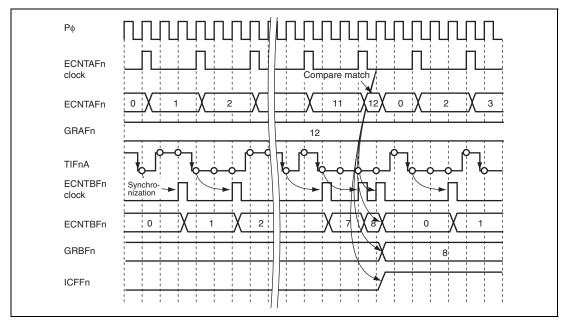


Figure 13.36 Operation Example of Edge Count in a Given Time

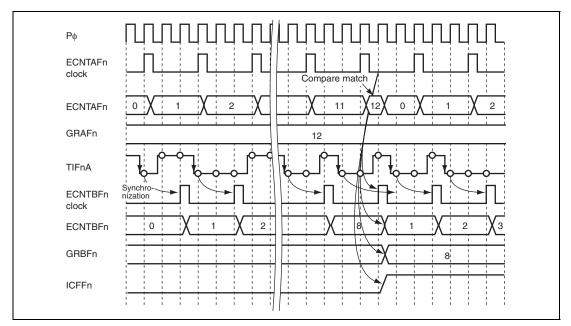


Figure 13.37 Operation Example of Edge Count in a Given Time (Compare Match and Event Occur Simultaneously)

## 13.26.2 Valid Edge Interval Counting

When a number of edges are set in GRBFn, the time necessary to count these edges is notified to GRAFn. The average of input edge intervals is obtained by dividing the time by the number of edges. The outcome is given as the unit of the ECNTAFn count source clock (GRAFn). Operation of the timer Fn is described below. Figure 13.38 shows an operation example. In this example, 13 cycles of the counter clock are needed to detect 12 input edges. Timer counter ECNTAFn and event counter ECNTBFn are driven by the ECNTAFn and ECNTBFn clocks, respectively.

Operation of registers in 'valid edge counting' mode is shown below.

- ECNTAFn: Measures time using one of the clock buses 0 to 5. When a compare match
  between ECNTBFn and GRBFn is detected, the count value is cleared synchronized with the
  next ECNTAFn clock. Since ECNTAFn count clear occurs at the same time with countup, the
  cleared value becomes H'00000100.
- ECNTBFn: Counts edges provided from TIFnA. Edge types subject to count can be selected from among rising, falling, or both edges. The example given here counts the falling edges. A delay of two cycles in TIFnA occurs because of synchronization processing. When a compare match is detected, the count number is cleared synchronized with the next Pφ clock.
- GRAFn: Functions as the capture register for ECNTAFn. When a compare match in ECNTBFn is detected, this register captures the ECNTAFn count number synchronizing with the next ECNTAFn clock.
- GRBFn: Functions as the compare match register for ECNTBFn. A compare match is detected when the count values in ECNTBFn and GRBFn agree.
- ICFFn flag: After detecting a compare match in ECNTBFn, sets the ICFFn flag synchronized with the next ECNTAFn clock.
- ECNTCFn, GRCFn, GRDFn: Do not function.

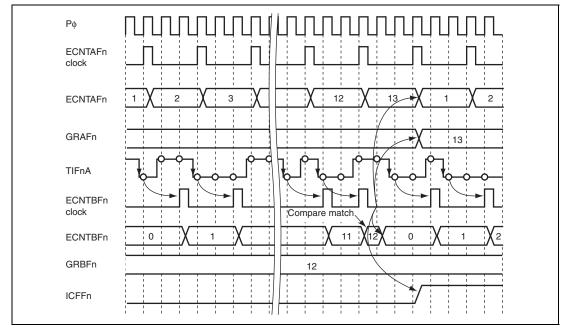


Figure 13.38 Operation Example of Valid Edge Interval Counting

# 13.26.3 Measurement of Time during High/Low Input Levels

Measures the time while TIFnA is driven high or low. The time obtained is indicated using the ECNTAFn clock source as the standard. The width of the measument time is specified to GRBFn in the form of the pulse number provided for TIFnA (GRBFn value). Operation of the timer F is described below. Figure 13.39 shows an operation example. This is the example in which the high level periods of the three pulses are measured as nine count source cycles. Timer counter ECNTAFn and event counter ECNTBFn are driven by the ECNTAFn and ECNTBFn clocks, respectively.

Operation of registers in 'counting high or low level of input' mode is shown below.

• ECNTAFn: Executes upcount using one of the clock buses 0 to 5 as a count source and TIFnA level as enable. Therefore, the time period in which TIFnA is in high level is measured. After detecting a compare match in ECNTBFn, this register clears the count number synchronizing with the next count source clock. If TIFnA is driven high at clearing count by the compare match, the count value becomes H'00000100. Figure 13.40 is an example of this.

- ECNTBFn: Counts the falling edge of TIFnA. A delay of two cycles occurs because of synchronization processing. When a compare match is detected, the count number is cleared synchronized with the next Pφ clock.
- GRAFn: Functions as the capture register for ECNTAFn. When a compare match in ECNTBFn is detected, this register captures the ECNTAFn count number synchronizing with the next ECNTAFn clock.
- GRBFn: Functions as the compare match register for ECNTBFn. A compare match is detected when the count number in ECNTBFn and GRBFn agree.
- ICFFn flag: After detecting a compare match in ECNTBFn, sets the ICFFn flag synchronized with the next ECNTAFn clock.
- ECNTCFn, GRCFn, GRDFn: Do not function.

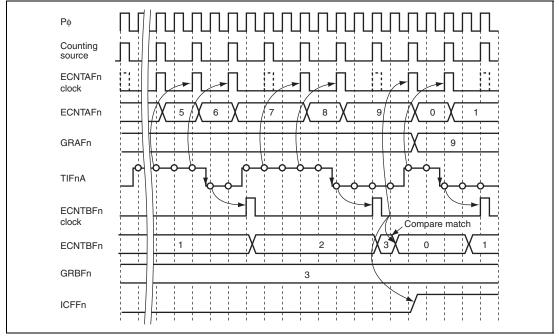


Figure 13.39 Operation Example of Measurement of Time during High Input Levels

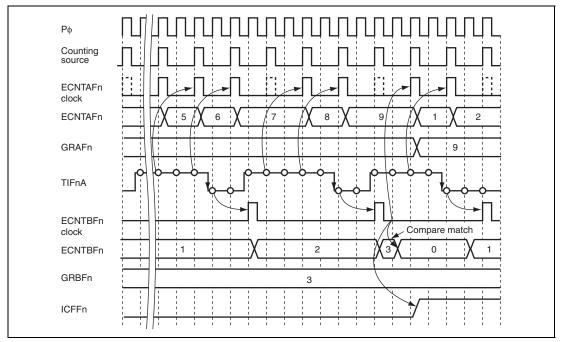


Figure 13.40 Operation Example of Measurement of Time during High Input Levels (TIFnA is in High Level When Capture is in Operation)

## 13.26.4 Measurement of PWM Input Waveform Timing

Measures the off-duty (non-active) period and cycle time of the PWM waveform input to TIFnA. The off-duty period is measured as the period of either the high or low level input on TIFnA, and the PWN cycle is measured as the interval between two rising or falling edges. Both are measured concurrently.

The measured time is expressed in the number of cycles of the clock source for ECNTAFn. The duration of the measuremt is set in GRBFn, which is specified as the number of PWM pulses input to TIFnA.

Operation of timer F is described below. Figure 13.41 shows an operation example. This is the example in which two PWM cycles in PWM waveform are measured as six counter clock cycles and the off-duty period (low-level period) is measured as four counter clock cycles.

The clocks for ECNTAFn, ECNTBFn, and ECNTCFn in this example provide the timing of counting or clearing operation of the time counter ECNTAFn, event counter ECNTBFn, and ECNTCFn, respectively.

The operation of each register in 'measurement of PWM input waveform timing' mode are as follows:

- ECNTAFn: Executes upcount using one of the clock buses 0 to 5 as a count source and TIFnA level as enable. Therefore, the time period in which TIFnA is in low level is measured. After detecting a compare match in ECNTBFn, this register clears the count number synchronizing with the next count source clock. If TIFnA is driven low at clearing count by the compare match, the count value becomes H'00000100.
- ECNTBFn: Counts the rising edge of TIFnA. A delay of two cycles occurs because of sybchronization processing. When a compare match is detected, the count number is cleared synchronized with the next Pφ clock.
- ECNTBFn: Counts the falling edge of TIFnA. A delay of two cycles occurs because of synchronization processing. When a compare match is detected, the count number is cleared synchronized with the next P\$\phi\$ clock.
- GRAFn: Functions as the capture register for ECNTAFn. When a compare match in ECNTBFn is detected, this register captures the ECNTAFn count number synchronizing with the next ECNTAFn clock.
- GRBFn: Functions as the compare match register for ECNTBFn. A compare match is detected when the count number in ECNTBFn and GRBFn agree.
- ECNTCFn: Measures time using the same count source as ECNTAFn. This register clears the
  count number synchronizing with the next ECNTAFn clock after detecting a compare match in
  ECNTBFn. Since ECNTCFn count clear occurs at the same time eith countup, the cleared
  value is H'00000100.
- GRCFn: Functions as the capture register for ECNTCFn. This register captures the ECNTCFn
  count number synchronizing with the next ECNTAFn clock after detecting a compare match in
  ECNTBFn.
- ICFFn flag: After detecting a compare match in ECNTBFn, sets the ICFFn flag synchronized with the next ECNTAFn clock.
- GRDFn: Does not function.

Therefore, ECNTBFn (GRBFn) and ECNTAFn (GRAFn) are operating in measurement of time during low input levels mode and ECNTBFn (GRBFn) and ECNTCFn (GRCFn) are operating in valid edge interval counting mode.

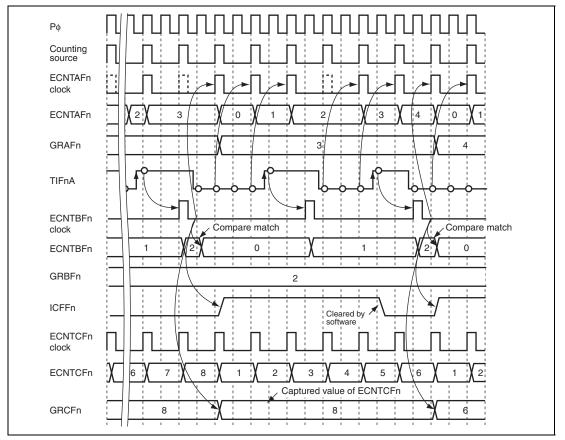


Figure 13.41 Operation Example of Measurement of PWM Input Waveform Timing

# 13.26.5 Rotation Speed/Pulse Measurement

Measures the number of edges input to TIFnA, the edge input time (time stamp), the off-duty period in the PWM waveform that emerges between the last input edge and the edge this time, and PWM cycle time.

The time obtained is expressed using the ECNTAFn clock source as the standard. The maximum interval of edge input can be set to GRBFn, which enables to output an interrupt request if the edge input interval exceeds the maximum value.

At this moment, the timer F operates as shown below. Figure 13.42 shows an example of operation. The ECNTAFn clock, ECNTBFn clock, and ECNTCFn clock show the timing of count

operation or clearing for the time counter ECNTAFn, event counter ECNTBFn, and ECNTCFn, respectively.

The operations of each register in 'rotation speed/pulse measurement' mode are as follows:

- ECNTAFn: Executes upcount using one of the clock buses 0 to 5 as a count source and TIFnA input level as enable. Therefore, the time period in which TIFnA is in low level is measured. After inputting the edge to TIFnA, this register clears the count number synchronizing with the next count source clock. If TIFnA is driven low at clearing count, the count value becomes H'00000100.
- ECNTBFn: Counts the rising edge of TIFnA. A delay of two cycles occurs because of sybchronization processing.
- GRAFn: Functions as the capture register for ECNTAFn. This register captures the ECNTAFn count number synchronizing with the next ECNTAFn clock after inputting the edge to TIFnA.
- GRBFn: Functions as the capture register for ECNTCFn. When the ECNTCFn count and the
  value in lower eight bits in GRBFn extended with 0 match, this register detects a compare
  match and set the OVFCFn to 1.
- ECNTCFn: Measures time using the same count source as ECNTAFn. This register clears the count number synchronizing with the next ECNTAFn clock after inputting the edge to TIFnA. Since ECNTCFn count clear occurs in the same timing, the cleared value is H'00000100.
- GRCFn: Functions as the capture register for ECNTCFn. This register captures the ECNTCFn count number synchronizing with the next ECNTAFn clock after inputting the edge to TIFnA.
- GRDFn: Functions as the capture register for ECNTCFn. This register captures the
  ECNTAFn, whose number being accumulated to GRDFn, synchronizing with the next
  ECNTAFn clock after inputting the edge to TIFnA. The value to be added is the ECNTCFn
  value before clearing.
- ICFFn flag: Sets the ICFFn flag synchronizing with the next ECNTAFn clock after inputting the edge to TIFnA.
- OVFCFn flag: Sets the OVFCFn synchronizing with the next Pφ clock after the values in ECNTCFn and GRBFn (in lower eight bits extended with 0) match.

While the ICFFn flag is set to 1, information on edge number, off duty cycle, PWM cycle, and edge input time can be obtained by reading ECNTBFn, GRAFn, GRCFn, and GRDFn, respectively. The capture timing of GRAFn, GRCFn, and GRDFn synchronizes with the count clock of ECNTAFn. Note that if the edge input cycle is shorter than the ECNTAFn count clock cycle, incorrect measurement may occur.

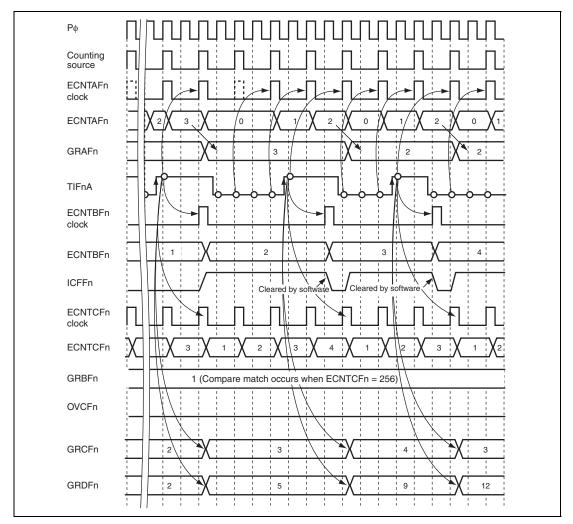


Figure 13.42 Operation Example of Rotation Speed/Pulse Measurement

## 13.26.6 Up/Down Event Count

This register uses the TIFnA pin, one of the two external input pins (TIFnA, TIFnB), as the count source, and TIFnB switches upcount to and from downcount. If a count period is designated to GRAFn, the count number after designation can be obtained in GRBFn. The counting period is the period of ECNTAFn count source clock (GRAFn value).

At this moment, the timer F operates as shown below. Figure 13.43 shows an example of operation. The ECNTAFn clock and ECNTBFn clock show the timing that time counter ECNTAFn and event counter ECNTBFn execute the count operation or clearing, respectively.

The operations of each register in up/down count operation mode are as follows:

- ECNTAFn: Measures time using one of the clock buses 0 to 5. When a compare match is detected, the count value is cleared synchronized with the next Pφ clock.
- ECNTBFn: Upcount/downcount operation is performed at both rising and falling edges of TIFnA. Count direction is detedmined by the TIFnB input level. (See table 13.16.) Because of synchronization processing, a delay of two cycles occurs in TIFnA and TIFnB.
- GRAFn: Functions as the compare match register for ECNTAFn. A compare match is detected when the count number in ECNTAFn and GRAFn agree.
- GRBFn: Functions as the capture register for ECNTBFn. When a compare match in ECNTAFn is detected, this register captures the ECNTBFn count number synchronizing with the next Pφ clock.
- ICFFn flag: After detecting a compare match in ECNTAFn, sets the ICFFn flag synchronized with the next Pφ clock.
- ECNTCFn, GRCFn, GRDFn: Do not function

Table 13.16 Count Direction in Up/Down Event Count Mode

|       | Count Direction |      |      |        |  |  |  |  |
|-------|-----------------|------|------|--------|--|--|--|--|
| Input | Upco            | ount | Dowr | ncount |  |  |  |  |
| TIFnA |                 | 7_   |      |        |  |  |  |  |
| TIFnB | Lo              | W    | H    | igh    |  |  |  |  |

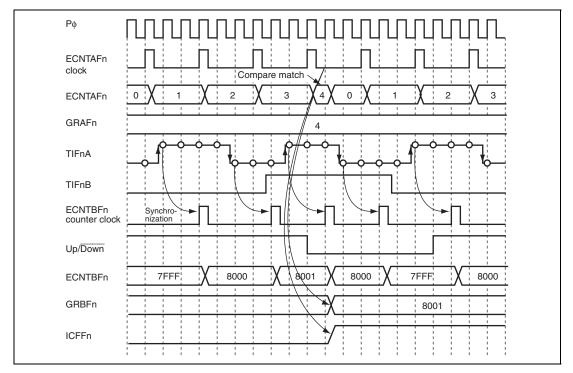


Figure 13.43 Operation Example of Up/Down Event Count

## 13.26.7 Four-time Multiplication Event Count

The count operation is executed using the external two input pins (TIFnA, TIFnB) as the count sources. Upcount or downcount is switched according to their input states. If a count period is designated to GRAFn, the count number after designation can be obtained in GRBFn. The counting period is the period of ECNTAFn cont source clock (GRAFn value).

At this moment, the timer F operates as shown below. Figure 13.44 shows an example of operation. The ECNTAFn clock and ECNTBFn clock show the timing that time counter ECNTAFn and event counter ECNTBFn execute the count operation or clearing, respectively.

The operations of each register in 'four-time event count operation' mode are as follows:

- ECNTAFn: Measures time using one of the clock buses 0 to 5. When a compare match is detected, the count value is cleared synchronized with the next Pφ clock.
- ECNTBFn: Upcount/downcount operation is performed at both rising and falling edges of TIFnA and TIFnB respectively. Count direction is detedmined by the other signal input level. (See table 13.17.) Because of synchronization processing, a delay of two cycles occurs in TIFnA and TIFnB.
- GRAFn: Functions as the compare match register for ECNTAFn. A compare match is detected when the count number in ECNTAFn and GRAFn agree.
- GRBFn: Functions as the capture register for ECNTBFn. When a compare match in ECNTAFn is detected, this register captures the ECNTBFn count number synchronizing with the next Pφ clock.
- ICFFn flag: After detecting a compare match in ECNTAFn, sets the ICFFn flag synchronized with the next Pφ clock.
- ECNTCFn, GRCFn, GRDFn: Do not function

**Table 13.17 Count Direction in Four-time Multiplication Event Count Mode** 

|       |      |      |       | Count    | Direction |     |          |         |
|-------|------|------|-------|----------|-----------|-----|----------|---------|
| Input |      | Up   | count |          |           | Dow | ncount   |         |
| TIFnA | High | 7_   | Low   | <u>_</u> | High      | 7   | Low      | <u></u> |
| TIFnB |      | High | 7     | Low      | 7         | Low | <u>_</u> | High    |

Note: Operation when edge inputs in TIFnA and TIFnB are detected simultaneously is not guaranteed. The interval between edge inputs in TIFnA and TIFnB must be at least 1.5 cycles (P $\phi$  clock).

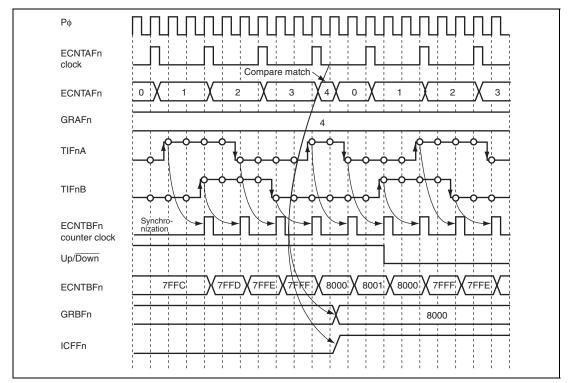


Figure 13.44 Operation Example of Four-time Multiplication Event Count Overflow/Underflow

When the count value H'FFFFF00 (ECNTAFn, ECNTCFn) changes to H'00000000 (ECNTAFn, ECNTCFn), or H'FFFF (ECNTBFn) to H'0000 (ECNTBFn) in other than counter clear processing, an overflow will be detected. In this case, a flag is immediately set with the value reaching H'00000000 (or H'0000). (No figure is given here.) OVFAF, ECNTBFn, or OVFCn are set when an overflow is detected in ECNTAFn, ECNTBF, or ECNTCFn, respectively.

An underflow will be detected when the value H'0000 (ECNTBFn) changes to H'FFFF (ECNTBFn) and a flag is immediately set with the value reaching H'FFFF. (No figure is given here.) An underflow occurs only in ECNTBFn and OVFBn is set.

### 13.26.8 Overflow and Underflow

When a counter value changes H'FFFFFF00 (ECNTAFn, ECNTCFn) to H'00000000 (ECNTAFn, ECNTCFn) and H'FFFF (ECNTBFn) to H'0000 (ECNTBFn) except counter clear operation, overflow is detected. Overflow flags are set at the same time when the counter value changes H'FFFFFF00 (or H'0000). OVFAFn, OVFBFn, and OVFCFn are set when overflow is detected in ECNTAFn, ECNTBFn, and ECNTCFn, respectively.

When a counter value changes H'0000 (ECNTBFn) to H'FFFF (ECNTBFn), underflow is detected. Underflow flag is set at the same time when the counter value changes from H'0000 to H'FFFF. Underflow occurs only in ECNTBFn, and OVFBFn is set upon its detection.

## 13.27 Overview of Timer G

Timer G consists of six subblocks which are identical with each other.

An active-low pulse is output for one cycle of the  $P\phi$  clock after a given time has elapsed. The time is counted. The generated pulse is used to activate the A/D converter or interrupt trigger.

Interrupt requests can be issued other than the pulse signal and can request DMA transfer to the DMAC.

The counter clock is selected from six lines of the clock bus.

## 13.27.1 Block Diagram of Robots

Timer G subblocks consist of one 16-bit timer counter G (TCNTG) and one compare match register (OCRG), and controller.

Figure 13.45 is a block diagram of timer G.

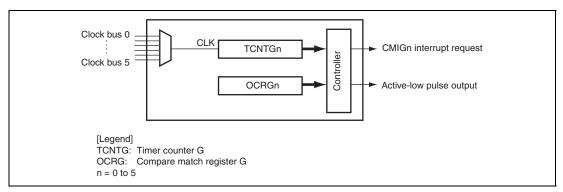


Figure 13.45 Block Diagram of Timer G

# 13.27.2 Interrupt Requests

Six timer G interrupts, CMIG0 to CMIG5, are available in timer G. When a compare match is detected in the subblocks, an interrupt request is output. The interrupt request is received in the direct memory access controller (DMAC) and interrupt controller (INTC).

# 13.28 Description of Timer G Registers

# 13.28.1 Timer Start Register G (TSTRG)

TSTRG is an 8-bit readable/writable register that enables and disables the subblocks of timer G. Timer G counters run when the STRG bit and the TGE bit in the ATU-III master enable register (ATUENR) are both set to 1.

TSTRG can be read from and written to in byte or word units.

TSTRG is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5     | 4 3   |       | 2     | 1     | 0     |  |
|----------------|---|---|-------|-------|-------|-------|-------|-------|--|
|                | - | - | STRG5 | STRG4 | STRG3 | STRG2 | STRG1 | STRG0 |  |
| Initial value: | 0 | 0 | 0     | 0     | 0     | 0     | 0     | 0     |  |
| R/W·           | R | R | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |  |

|      |          | Initial |     |   |
|------|----------|---------|-----|---|
| Bit  | Bit Name | Value   | R/W | Description   |
| 7, 6 | _        | All 0   | R   | Reserved  |
|      |          |         |     | These bits are always read as 0. The write value should always be 0.  |
| 5    | STRG5    | 0       | R/W | Counter G Start   |
| 4    | STRG4    | 0       | R/W | These bits enable and disable timer counter Gn (TCNTGn)   |
| 3    | STRG3    | 0       | R/W | in the subblock.  |
| 2    | STRG2    | 0       | R/W | <ul> <li>When these bits are cleared to 0, TCNTGn is stopped.</li> <li>While TCNTGn is stopped, it retains the previous value.</li> </ul>   |
| 1    | STRG1    | 0       | R/W | When these bits are set to 1, TCNTGn is resumed from the  |
| 0    | STRG0    | 0       | R/W | revious value. TCNTGn runs when these bits and the TGE bit in ATUENR are both set to 1.   |
|      |          |         |     | 0: TCNTGn is disabled   |
|      |          |         |     | 1: TCNTGn is enabled  |
|      |          |         |     | The prescalers run regardless of the counter G start bit and are not synchronized with the timing at which TCNTG is started. Therefore, the time from when the counter G start bit is set to when TCNTG is incremented for the first time is less than the cycle of the clock of TCNTG. |

Note: n = 0 to 5 (n denotes subblock number)

# 13.28.2 Timer Control Register G0 to G5 (TCRG0 to TCRG5)

TCRG0 to TCRG5 are 8-bit readable/writable registers that set the operating mode of each subblock of timer G.

TCRG0 to TCRG5 can be read from and written to in byte or word units.

TCRG0 to TCRG5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 5 |        | 4     | 3 | 2 | 1           | 0         |
|----------------|---|-----|--------|-------|---|---|-------------|-----------|
|                | - | CKS | SELGn[ | [2:0] | - | - | CMP<br>OEGn | CM<br>EGn |
| Initial value: | 0 | 0   | 0      | 0     | 0 | 0 | 0           | 0         |
| R/W:           | R | R/W | R/W    | R/W   | R | R | R/W         | R/W       |

|        |          | Initial |     |  |
|--------|----------|---------|-----|--|
| Bit    | Bit Name | Value   | R/W | Description  |
| 7      | _        | 0       | R   | Reserved   |
|        |          |         |     | This bit is always read as 0. The write value should always be 0.  |
| 6 to 4 | CKSELGn  | 000     | R/W | Clock Select Gn  |
|        | [2:0]    |         |     | These bits select the clock source of timer counter Gn (TCNTGn) of subblock. However, do not set to B'110 or B'111. If set, operation cannot guaranteed. |
|        |          |         |     | 000: Clock-bus line 0  |
|        |          |         |     | 001: Clock-bus line 1  |
|        |          |         |     | 010: Clock-bus line 2  |
|        |          |         |     | 011: Clock-bus line 3  |
|        |          |         |     | 100: Clock-bus line 4  |
|        |          |         |     | 101: Clock-bus line 5  |
|        |          |         |     | 110: Setting prohibited  |
|        |          |         |     | 111: Setting prohibited  |
| 3, 2   | _        | All 0   | R   | Reserved   |
|        |          |         |     | These bits are always read as 0. The write value should always be 0.   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 1   | CMPOEGn  | 0                | R/W | Pulse Output Enable Gn   |
|     |          |                  |     | Selects whether or not a compare match pulse is externally output on compare match between timer counter Gn (TCNTGn) and compare match register (OCRGn). |
|     |          |                  |     | Pulse is not output on compare match between<br>TCNTGn and OCRGn   |
|     |          |                  |     | Pulse is output on compare match between TCNTGn and OCRGn  |
| 0   | CMEGn    | 0                | R/W | Compare Match Interrupt Enable Gn  |
|     |          |                  |     | Enables and disables output of interrupt requests on compare match between timer status register Gn (TSRGn) and compare match flag Gn (CMFGn).           |
|     |          |                  |     | 0: Interrupt request is not issued on compare match of CMFGn   |
|     |          |                  |     | 1: Interrupt request is issued on compare match of CMFGn   |

## 13.28.3 Timer Status Registers G0 to G5 (TSRG0 to TSRG5)

TSRG0 to TSRG5 are 8-bit readable/writable registers that measure the time and indicate occurrence of event counter overflow and compare match.

These flags are interrupt request sources and interrupt requests for the CPU or DMA transfer requests for the DMAC can be issued when bits in timer control registers G0 to G5 (TSRG0 to TSRG5) are set to 1.

TSRG0 to TSRG5 can be read from and written to in byte or word units.

TSRG0 to TSRG5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1      | 0      |
|----------------|---|---|---|---|---|---|--------|--------|
|                | 1 | - | ı | - | - | 1 | OVFGn  | CMFGn  |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0      | 0      |
| R/W:           | R | R | R | R | R | R | R/(W)* | R/(W)* |

<sup>\*</sup> Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

| Bit    | Bit Name | Initial<br>Value | R/W    | Description   |
|--------|----------|------------------|--------|---|
| 7 to 2 | _        | All 0            | R      | Reserved  |
|        |          |                  |        | These bits are always read as 0. The write value should always be 0.  |
| 1      | OVFGn    | 0                | R/(W)* | Overflow Flag Gn  |
|        |          |                  |        | Indicates whether or not timer counter Gn (TCNTGn) has overflowed. This bit cannot be set to 1 by software. No interrupt corresponds to this bit. |
|        |          |                  |        | 0: TCNTGn has not overflowed  |
|        |          |                  |        | [Clearing condition]  |
|        |          |                  |        | • When writing 0 to this bit after reading it as 1  |
|        |          |                  |        | 1: TCNTGn has overflowed  |
|        |          |                  |        | [Setting condition]   |
|        |          |                  |        | When TCNTGn overflowed (from H'FFFF to H'0000)  |
| 0      | CMFGn    | 0                | R/(W)* | Compare Match Flag Gn   |
|        |          |                  |        | Indicates whether or not a compare match has occurred in subblocks G0 to G5. This flag cannot be set to 1 by software.                            |
|        |          |                  |        | When the CMEGn bit in the timer control register is set to 1 and this flag is set to 1, a compare match interrupt is issued.                      |
|        |          |                  |        | 0: Compare match has not occurred in subblocks Gn   |
|        |          |                  |        | [Clearing conditions]   |
|        |          |                  |        | • When writing 0 to this bit after reading it as 1  |
|        |          |                  |        | <ul> <li>When DMAC received transfer request by compare match interrupt</li> </ul>  |
|        |          |                  |        | 1: Compare match has occurred in subblocks Gn   |
|        |          |                  |        | [Setting condition]   |
|        |          |                  |        | When compare match occurred in subblocks Gn   |

## 13.28.4 Timer Counters G0 to G5 (TCNTG0 to TCNTG5)

TCNTG0 to TCNTG5 are 16-bit readable/writable registers. These registers are provided one for each subblock and are incremented by the clock selected in the corresponding control register. Lines 0 to 5 of the clock bus can be selected as the input clock.

These counter values are constantly compared with the value in compare match register G (OCRG). When they match, Compare match flag G (CMFG) is set and the counter value is cleared to H'0000 in the next P $\phi$  clock cycle. If counter clearing by compare match and incrementation occur simultaneously, TCNTG is initialized to H'0001. This occurs when TCNTG is driven by the clock whose frequency is equal to the P $\phi$  clock.

TCNTG0 to TCNTG5 can be read from and written to in word units.

TCNTG0 to TCNTG5 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

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

| Bit     | Bit Name | Initial<br>Value | R/W | Description                            |
|---------|----------|------------------|-----|--|
| 15 to 0 |          | All 0            | R/W | Timer Counter Gn                       |
|         | [15:0]   |                  |     | These bits store the up-counter value. |

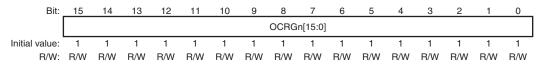
# 13.28.5 Compare Match Registers G0 to G5 (OCRG0 to OCRG5)

OCRG0 to OCRG5 are 16-bit readable/writable registers. These registers are provided one for each subblock and function as the output compare register for timer counter G (TCNTG).

Do not set OCRG to H'0000. If H'0000 is set, compare matches occur at unwanted cycles.

OCRG0 to OCRG5 can be read from and written to in word units.

OCRG0 to OCRG5 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.



| Bit      | Bit Name  | Initial<br>Value | R/W   | Description                             |
|----------|-----------|------------------|-------|---|
| <u> </u> | Dit Haine | Value            | 10/11 | Description                             |
| 15 to 0  | OCRGn     | All 1            | R/W   | Compare Match Gn                        |
|          | [15:0]    |                  |       | These bits set the compare match value. |

# 13.29 Operations of Timer G

An active-low pulse is output for one cycle of the  $P\phi$  clock after a time set in OCRG has elapsed. The initial level on the output pin is a level of 1. Set the number of cycles of the TCNTG clock in OCRG.

The generated pulse can be used to activate the A/D converter by setting the compare match pulse output enable bit (CMPOEG) in timer control register G (TCRG).

When compare match occurs, the compare match flag (CMFG) in timer status register G (TSRG) is set. DMA transfer or interrupt requests can be issued for the DMAC or CPU by setting the compare match interrupt enable bit (CMEG) in TCRG.

Figure 13.46 shows operation example for counters and compare match. In this example, the TCNTG clock is an ideal signal to show count timing or clearing timing.

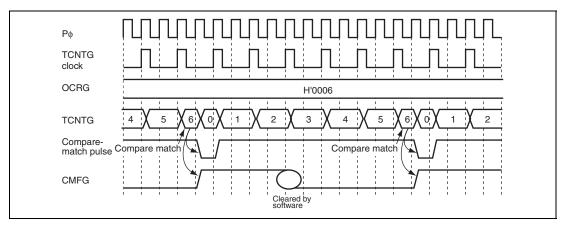


Figure 13.46 Operation Example of Counter and Compare Match

#### 13.30 Overview of Timer H

Timer H is a counter that measures a given time repeatedly.

Timer counter 1H (TCNT1H) is a 16-bit up-counter driven by the clock selected from six lines of the clock bus. When the value in TCNT1H reaches the value in compare register 1H (OCR1H), it is cleared to H'0000 and is incremented again.

Timer counter 2H (TCNT2H) is a 32-bit counter that is incremented on compare match between TCNT1H and OCR1H, meaning that the counter counts the compare match.

Interrupt requests can be issued when TCNT2H is incremented.

#### 13.30.1 **Block Diagram of Timer H**

Timer H consists of one 16-bit timer counter 1H (TCNT1H), one compare match register 1H (OCR1H), one 32-bit timer counter 2H (TCNT2H), and controller.

Figure 13.47 is a block diagram of timer H.

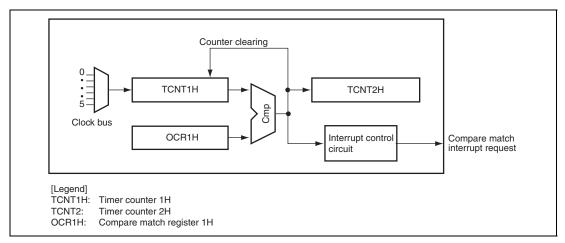


Figure 13.47 Block Diagram of Timer H

# **13.30.2 Interrupts**

Timer H has one interrupt request signal (CMIH). Interrupt requests are issued on compare match between TCNT1H and OCR1H.

# 13.31 Description of Timer H Registers

# 13.31.1 Timer Control Register H (TCRH)

TCRH is an 8-bit readable/writable register that selects the counter clock and controls output of compare match interrupt.

TCRH can be read from and written to in byte or word units.

TCRH is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 5 |        | 4    | 3 | 2 | 1 | 0    |
|----------------|---|-----|--------|------|---|---|---|------|
|                | - | СК  | SELH[2 | 2:0] | - | - | - | СМЕН |
| Initial value: | 0 | 0   | 0      | 0    | 0 | 0 | 0 | 0    |
| R/W:           | R | R/W | R/W    | R/W  | R | R | R | R/W  |

|        |          | Initial |     |   |
|--------|----------|---------|-----|---|
| Bit    | Bit Name | Value   | R/W | Description   |
| 7      | _        | 0       | R   | Reserved  |
|        |          |         |     | This bit is always read as 0. The write value should always be 0.       |
| 6 to 4 | CKSELH   | 000     | R/W | Clock Select H  |
|        | [2:0]    |         |     | These bits select the source of the clock of timer counter 1H (TCNT1H). |
|        |          |         |     | 000: Clock-bus line 0   |
|        |          |         |     | 001: Clock-bus line 1   |
|        |          |         |     | 010: Clock-bus line 2   |
|        |          |         |     | 011: Clock-bus line 3   |
|        |          |         |     | 100: Clock-bus line 4   |
|        |          |         |     | 101: Clock-bus line 5   |
|        |          |         |     | 110: Setting prohibited   |
|        |          |         |     | 111: Setting prohibited   |
| 3 to 1 | _        | All 0   | R   | Reserved  |
|        |          |         |     | These bits are always read as 0. The write value should always be 0.    |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 0   | CMEH     | 0                | R/W | Compare Match Interrupt Enable H  |
|     |          |                  |     | Enables and disables the interrupt request for compare match flag H (CMFH) in timer status register H (TSRH). |
|     |          |                  |     | 0: CMFH interrupt is disabled   |
|     |          |                  |     | 1: CMFH interrupt is enabled  |

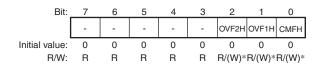
### Timer Status Register H (TSRH) 13.31.2

TSRH is an 8-bit readable/writable register that indicates occurrence of compare match between timer counter 1H (TCNT1H) and the compare match register (OCR1H), TCNT1H overflow, and overflow on timer counter 2H (TCNT2H).

The compare match flag is an interrupt source and can be used for output of the interrupt request when the bit in timer control register F (TCRH) is set to 1.

TSRH can be read from and written to in byte or word units.

TSRH is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Note: \* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 3 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial<br>Value | R/W              | Description  |
|-----|----------|------------------|------------------|--|
| 2   | OVF2H    | 0                |                  | Overflow Flag 2H   |
| ۷   | OVEZII   | U                | ⊓/( <b>vv</b> )· | Indicates whether or not timer counter 2H (TCNT2H) has overflowed. This flag cannot be set to 1 by software. No interrupt corresponds to this flag.  |
|     |          |                  |                  | 0: TCNT2H has not overflowed   |
|     |          |                  |                  | [Clearing condition]   |
|     |          |                  |                  | When writing 0 to this bit after reading it as 1   |
|     |          |                  |                  | 1: TCNT2H has overflowed   |
|     |          |                  |                  | [Setting condition]  |
|     |          |                  |                  | <ul> <li>TCNT2H has overflowed (from H'FFFFFFF to<br/>H'0000000)</li> </ul>  |
|     |          |                  |                  | Writing H'00000000 to TCNT2H or starting TCNT2H up from the initial value (H'00000000) has no effect on this bit   |
|     |          |                  |                  | When writing a value to the counter and incrementation occur simultaneously while the counter is H'FFFFFFF, the overflow flag is set to 1 but the counter value is changed to the written value instead of H'0000. |
| 1   | OVF1H    | 0                | R/(W)*           | Overflow Flag 1H   |
|     |          |                  |                  | Indicates whether or not timer counter 1H (TCNT1H) has overflowed. This flag cannot be set to 1 by software. No interrupt corresponds to this flag.  |
|     |          |                  |                  | 0: TCNT1H has not overflowed   |
|     |          |                  |                  | [Clearing condition]   |
|     |          |                  |                  | When writing 0 to this bit after reading it as 1   |
|     |          |                  |                  | 1: TCNT1H has overflowed   |
|     |          |                  |                  | [Setting condition]  |
|     |          |                  |                  | TCNT1H has overflowed (from H'FFFF to H'0000)  |
|     |          |                  |                  | Writing H'0000 to TCNT1H or starting TCNT1H up from the initial value (H'0000) has no effect on this bit   |
|     |          |                  |                  | When writing a value to the counter and incrementation occur simultaneously while the counter is H'FFFF, the overflow flag is set to 1 but the counter value is changed to the written value instead of H'0000.    |

| Bit | Bit Name | Initial<br>Value | R/W    | Description  |
|-----|----------|------------------|--------|--|
| 0   | CMFH     | 0                | R/(W)* | Compare Match Flag H   |
|     |          |                  |        | Indicates whether or not compare match between TCNT1H and OCR1H has occurred. This flag cannot be set to 1 by software. When the CMEH bit in TCRH is set to 1, a compare match interrupt is issued by setting this flag. |
|     |          |                  |        | 0: Compare match between TCNT1H and OCR1H has not occurred   |
|     |          |                  |        | [Clearing condition]   |
|     |          |                  |        | When writing 0 to this bit after reading it as 1   |
|     |          |                  |        | Compare match between TCNT1H and OCR1H has occurred  |
|     |          |                  |        | [Setting condition]  |
|     |          |                  |        | When the values in TCNT1H and OCR1H match  |
|     |          |                  |        | While CMFH is set to 1, meaning that the flag has not been cleared, the next compare match can be detected. In this case, 1 is rewritten to this bit.  |

Note: \* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

# 13.31.3 Timer Counter 1H (TCNT1H)

TCNT1H is a 16-bit readable/writable register. TCNT1H run when the THE bit in the ATU-III master enable register (ATUENR) and is incremented by the clock selected in the CKSEL bit in timer control register H (TCRH).

Overflow flag 1H (OVF1H) in timer status register H (TSRH) is set to 1 when TCNT1H overflows (from H'FFFF to H'0000)

The value in TCNT1H is constantly compared with the value in compare match register 1H (OCR1H). When they match, the CMFH bit in TSRH is set to 1 and TCNT1H is cleared to H'0000 in the next  $P\phi$  clock cycle.

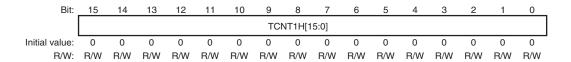
However, if counter clearing by compare match and incrementation occur simultaneously, TCNT1H is cleared to H'0001. This occurs when TCNT1H is driven by the clock whose frequency is equal to the  $P\phi$  clock.

At the same time as compare match, timer counter 2H (TCNT2H) is incremented.

Apr 01, 2014

TCNT1H can be read from and written to in word units.

TCNT1H is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.



| Bit     | Bit Name           | Initial<br>Value | R/W | Description                                   |
|---------|--------------------|------------------|-----|---|
| 15 to 0 | 0 TCNT1H All 0 R/W |                  | R/W | Timer Counter 1H                              |
|         | [15:0]             |                  |     | These bits store the 16-bit up-counter value. |

### **Compare Match Register 1H (OCR1H)** 13.31.4

OCR1H is a 16-bit readable/writable register that function as the output compare register for timer counter 1H (TCNT1H). Compare match occurs at the following cycles.

Cycle of compare match =

cycle of the TCNT1H counter clock (selected in the CKSELH bit in TCRH) × value in OCR1H

TCNT2H is incremented at this cycle. Interrupt requests are output when they are enabled,

Do not set OCR1H to H'0000. If set, compare match occurs at unwanted cycles.

OCR1H can be read from and written to in word units.

OCR1H is initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 15  | 14          | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|----------------|-----|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     | OCR1H[15:0] |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 1   | 1           | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| ₽/M·           | D/M | D/M         | D/M | D/W | D/M | D/M | D/M | D/M | D/M | D/M | RΛΜ | D/M | D/M | D/M | D/M | D/M |

| Bit     | Bit Name | Initial<br>Value | R/W | Description                               |
|---------|----------|------------------|-----|---|
| 15 to 0 | OCR1H    | H All 1 R/W      |     | Compare Match 1H                          |
|         | [15:0]   |                  |     | These bits store the compare match value. |

## 13.31.5 Timer Counter 2H (TCNT2H)

TCNT2H is a 32-bit readable/writable register. TCNT2H is incremented every compare match between timer counter 1H (TCNT1H) and compare match register 1H (OCR1H).

TCNT2H runs when the THE bit in the ATU-III master enable register (ATUENR).

When TCNT2H overflows (from H'FFFF FFFF to H'0000 0000), overflow flag 2H (OVF2H) in timer status register H (TSRH) is set to 1.

TCNT2H can be read from and written to in longword units.

TCNT2H is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 31            | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                | TCNT2H[31:16] |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 0             | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W           | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit:           | 15            | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                | TCNT2H[15:0]  |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | 0             | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R/W           | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit     | Bit Name | Initial<br>Value | R/W | Description                                |
|---------|----------|------------------|-----|--|
| 31 to 0 | TCNT2H   | TCNT2H All 0 R/W |     | Timer Counter 2H                           |
|         | [31:0]   |                  |     | These bits store the 32-bit counter value. |

# 13.32 Operations of Timer H

Compare match occurs and the compare match flag (CMFH) is set to 1 when a time set in the compare match register (OCR1H) has elapsed. At this time, TCNT2H is incremented and TCNT1H is cleared to H'0000.

The TCNT1H counter clock is selected by the CKSELH bit in TCRH. TCNT1H and TCNT2H run when the THE bit in ATUENR is set to 1. If the THE bit is cleared to 0 while the counters are in operation, TCNT1H and TCNT2H are stopped and retain the counter value unchanged. When the THE bit is set to 1, the counters are resumed from the retained value.

Interrupt requests can be issued depending on the compare match interrupt enable H bit (CMEH).

Figure 13.48 shows an operation example of timer H. In this example, the TCNT1H counter clock is an ideal signal to show counter operation or clearing operation timing.

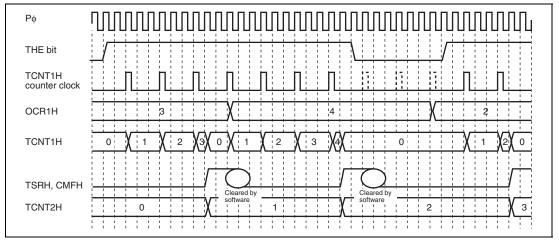


Figure 13.48 Operations of Timer H

### 13.33 Overview of Timer J

Timer J consists of two identical subblocks that measure a given time repeatedly.

Timer counter J (TCNTJ) is a 16-bit counter and is incremented by the clock selected from six lines of the clock bus. TCNTJ is cleared by the input edge of the TIJ pin. Timer J also has nine 16-bit FIFO registers which latch the value in TCNTJ every edge input. When the FIFO becomes full, DMAC activation or an interrupt request can be issued.

Timer J controls the valid period of the FIFO by controlling the time from compare match of compare match register J (OCRJ) to FIFO full.

Noise of the externlly input signals (TIJ) can be removed by the input noise cancelltion function.

## 13.33.1 Block Diagram of Timer J

Timer J consists of two identical subblocks. Each subblock includes one 16-bit timer counter J (TCNTJ), one compare match register J (OCRJ), nine 16-bit FIFO registers, input signal controller (edge extractors and noise cancelers), and controller.

Figure 13.49 is a block diagram of timer J.

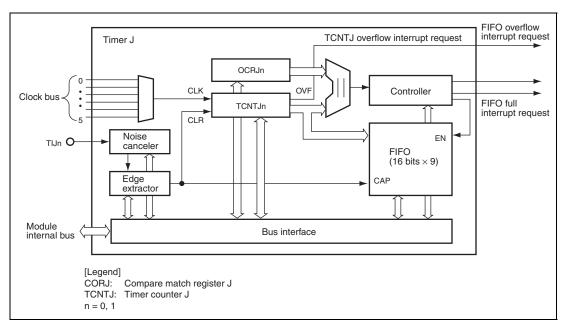


Figure 13.49 Block Diagram of Timer J

# 13.34 Description of Timer J Registers

## 13.34.1 Timer Start Register J (TSTRJ)

TSTRJ is an 8-bit readable/writable register that enables and disables the subblocks (timer J0 and timer J1) of timer J. Timer J counters run when the counter J start bit (STRJ) and timer J enable bit (TJE) in the ATU-III master enable register (ATUENR) are both set to 1.

The prescalers run regardless of the counter J start bit and are not sychronized with the timing at which the TCNTJn is started. Therefore, the time from when the THE bit is set to when TCNTJn is incremented for the first time is less than the cycle of the clock of TCNTJn.

TSTRJ can be read from and written to in byte or word units.

TSTRJ is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |
|----------------|---|---|---|---|---|---|-------|-------|
|                | - | - | - | - | - | - | STRJ1 | STRJ0 |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0     | 0     |
| R/W:           | R | R | R | R | R | R | R/W   | R/W   |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 2 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 1      | STRJ1    | 0                | R/W | Counter Jn Start   |
| 0      | STRJ0    | 0                | R/W | These bits enable and disable timer counter Jn (TCNTJn).   |
|        |          |                  |     | When these bits are cleared to 0, TCNTJn is stopped. While TCNTJn is stopped, it retains the previous value. When these bits are set again, the counter is resumed from the value. |
|        |          |                  |     | 0: Counting of TCNTJn is stopped   |
|        |          |                  |     | 1: Counting of TCNTJn is enabled   |

Note: n = 0.1

# 13.34.2 Timer Control Registers J0 and J1 (TCRJ0 and TCRJ1)

TCRJ0 and TCRJ1 are 8-bit readable/writable registers that select operation modes of each subblock (timer J0 and timer J1)

TCRJ0 and TCRJ1 can be read from and written to in byte or word units.

TCRJ0 and TCRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6   | 5      | 4    | 3 | 2     | 1    | 0     |
|----------------|---|-----|--------|------|---|-------|------|-------|
|                | - | CK  | SELJn[ | 2:0] | - | NCEJn | IOJn | [1:0] |
| Initial value: | 0 | 0   | 0      | 0    | 0 | 0     | 0    | 0     |
| R/W:           | R | R/W | R/W    | R/W  | R | R/W   | R/W  | R/W   |
|                |   |     |        |      |   |       |      |       |

Note: n = 0, 1

| Bit    | Bit Name | Initial<br>Value | R/W  | Description   |
|--------|----------|------------------|--|---|
| 7      | _        | 0                | R  | Reserved  |
|        |          |                  |  | This bit is always read as 0. The write value should always be 0. |
| 6 to 4 | CKSELJn  | 000              | R/W  | Clock Select Jn   |
|        | [2:0]    |                  | These bits select the clock source of timer counter Jn (TCNTJn) of subblock. However, do not set to B'110 or B'111. If set, operation cannot guaranteed. |   |
|        |          |                  |  | 000: Clock-bus line 0   |
|        |          |                  |  | 001: Clock-bus line 1   |
|        |          |                  |  | 010: Clock-bus line 2   |
|        |          |                  |  | 011: Clock-bus line 3   |
|        |          |                  |  | 100: Clock-bus line 4   |
|        |          |                  |  | 101: Clock-bus line 5   |
|        |          |                  |  | 110: Setting prohibited   |
|        |          |                  |  | 111: Setting prohibited   |
| 3      | _        | 0                | R  | Reserved  |
|        |          |                  |  | This bit is always read as 0. The write value should always be 0. |

| D:4 | Dit Name | Initial | D/M | Description   |
|-----|----------|---------|-----|---|
| Bit | Bit Name |         | R/W | Description   |
| 2   | NCEJn    | 0       | R/W | Noise Canceler Enable Jn  |
|     |          |         |     | These bits enable and disable the noise cancelers for externally input signals (TIJn)   |
|     |          |         |     | When a level change on externally input signal TIJn is detected while this bit is set to 1, it is processed in premature-transition cancellation or minumum time-at-level cancellation mode depending on the setting in the noise cancelltion mode register (NCMR) of the common controller.  |
|     |          |         |     | In premature-transition cancellation mode   |
|     |          |         |     | When a level change of the externally input signal is detected, the change is output as the signal whose noise is removed and the corresponding noise canceler counter (NCNTJn) is started for counting up. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRJn). The level of the externally input signal is output on this compare match.  |
|     |          |         |     | When these bits are cleared to 0 while the counter (NCNTJn) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.  |
|     |          |         |     | In minumum time-at-level cancellation mode  |
|     |          |         |     | When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTJn) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRJn), the previously accepted level change is output as the siganl whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed. |
|     |          |         |     | When these bits are cleared to 0 while the counter (NCNTJn) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected.  |
|     |          |         |     | For details on operations in each mode, see figures 13.1 and 13.2.  |
|     |          |         |     | 0: Noise canceler for TIJn is disabled  |
|     |          |         |     |   |

1: Noise canceler for TIJn is enabled

|      |           | Initial |     |   |
|------|-----------|---------|-----|---|
| Bit  | Bit Name  | Value   | R/W | Description   |
| 1, 0 | IOJn[1:0] | 00      | R/W | I/O Control Jn  |
|      |           |         |     | These bits select the edge of external inputs (TIJn) that is to be extracted for use in input-capture triggering. When an edge selected in these bits is detected, the contents of timer counter Jn (TCNTNJn) are transferred to FIFO J (FIFOJn). |
|      |           |         |     | Edges are extracted from the signal affter noise removal.   |
|      |           |         |     | When the noise canceler is disabled, the selected edge is simply extracted from the external inputs (TIJn). When the noise canceler is enabled, the selected edge is extracted from the signals after noise removal.                              |
|      |           |         |     | Edge extraction is synchronized with the $P\varphi$ clock. Make sure that the frequency of the $P\varphi$ clock is at least twice the frequency of the external input signal. Otherwise, edge extraction will not performed correctly.            |
|      |           |         |     | 00: Input capturing is not performed  |
|      |           |         |     | 01: TCNTJ is captured in ICRJ on the rising edge of TIJ   |
|      |           |         |     | 10: TCNTJ is captured in ICRJ on the falling edge of TIJ  |
|      |           |         |     | 11: TCNTJ is captured in ICRJ on both edges of TIJ  |

Note: n = 0, 1

# 13.34.3 FIFO Control Registers J0 and J1 (FCRJ0 and FCRJ1)

FCRJ0 and FCRJ1 are 8-bit readable/writable registers that control FIFO of each subblock (timer J0 and timer J1).

FCRJ0 and FCRJ1 can be read from and written to in byte or word units.

FCRJ0 and FCRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7            | 6 | 5            | 4          | 3 | 2 | 1          | 0           |
|----------------|--------------|---|--------------|------------|---|---|------------|-------------|
|                | FIFO<br>ENJn | - | FVCR<br>ENJn | FRS<br>TJn | 1 | - | FDF<br>GJn | TR<br>[1:0] |
| Initial value: | 0            | 0 | 0            | 0*         | 0 | 0 | 0          | 0           |
| R/W:           | R/W          | R | R/W          | W          | R | R | R/W        | R/W         |

Note: n = 0, 1

<sup>\*</sup> Writing 0 is ignored. This bit is always read as 0 even if 1 is written.

|     |          | Initial |     |  |
|-----|----------|---------|-----|--|
| Bit | Bit Name | Value   | R/W | Description  |
| 7   | FIFOENJn | 0       | R/W | FIFO Register Enable Jn  |
|     |          |         |     | Enables and disables FIFO register Jn (FIFOJn) in subblocks J0 and J1. When this bit is set to 1 and the FIFO is disabled, even if an edge is input on pin TIJn, the value in TCNTJn is not latched in FIFO. However when FVCRENJn is set to 1, even if this bit is 0, the value in TCNTJn is latched in the FIFO every input edge until FIFO becomes full after compare match on OCRJn. |
|     |          |         |     | 0: FIFO Jn is disabled   |
|     |          |         |     | 1: FIFO Jn is enabled  |
| 6   | _        | 0       | R   | Reserved   |
|     |          |         |     | This bit is always read as 0. The write value should always be 0.  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 5    | FVCRENJn | 0                | R/W | FIFO Enable Control Enable Jn   |
|      |          |                  |     | When FIFOENJn = 0 and the FIFO is disabled, the FIFO is temporarily enabled until it becomes full (the threshold level can be set by FDFTRGJ) after compare match of compare match register Jn (OCRJn). Note that the FIFO is always enabled regardless of this bit when bit FIFOENJn is set to 1.  |
|      |          |                  |     | 0: FIFO is not enabled on OCRJn compare match   |
|      |          |                  |     | FIFO is enabled until it becomes full after OCRJn compare match   |
| 4    | FRSTJn   | 0*1              | W   | FIFO Data Register Reset Jn   |
|      |          |                  |     | Data captured in the FIFO is discarded and then the FIFO becomes empty (FIFO reset). However, the FDFFJn and FDOVFJn flags which have been set to 1 are not cleared.  |
|      |          |                  |     | 0: No operation   |
|      |          |                  |     | 1: FIFO is reset  |
| 3, 2 | _        | All 0            | R   | Reserved  |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 1, 0 | FDFTRGJn | 00               | R/W | FIFO Data Full Trigger Jn   |
|      | [1:0]    |                  |     | These bits set the number of data words which is the threshold for setting the FDFFJn flag in the timer J status register (TSRJn). When the FIFO is enabled and the number of data words in it has reached the trigger count, the FDFFJn bit is set to 1. When the FIFO is enabled on compare match while FIFOENJn = 0 and FVCRENJn = 1, and then the number of data words has reached the trigger count, the FIFO is disabled. |
|      |          |                  |     | 00: 9   |
|      |          |                  |     | 01: 6   |
|      |          |                  |     | 10: 4   |
|      | - 0.1    |                  |     | 11: 2   |

Notes: n = 0, 1

- 1. Writing 0 to this bit is ignored. The write data is not held. This bit is always read as 0.
- When the setting of FIFOENJn is 0 and that of FVCRENJn is 1, FDFTRGJn[1:0] can
  only be set to 00 when a FIFO data full interrupt is generated for the CPU. After
  generation of the FIFO data full interrupt, the service routine should read all data from
  the FIFO, use the FRSTJn bit to reset the FIFO registers, and then clear FDFFJn to 0.

## 13.34.4 Timer Status Register J0 and J1 (TSRJ0 and TSRJ1)

TSRJ0 and TSRJ1 are 8-bit readable/writable registers that indicate occurrence of overflow on timer counter Jn (TCNTJn), and compare match on compare register Jn (OCRJn). In addition, there are flags that indicate that the FIFO has overflowed and the number of data in the FIFO exceeds the trigger count.

The flags other that FIFO status flag Jn (FVLDFJn) and compare match flag Jn (CMFJn) are used for generating interrupt requests. When the corresponding bit in timer interrupt enable register Jn (TIERJn) is set to 1, DMA transfer requests for the DMAC or interrupt request for the CPU can be issued.

TSRJ0 and TSRJ1 can be read from and written to in byte or word units.

TSRJ0 and TSRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4           | 3         | 2      | 1           | 0          |
|----------------|---|---|---|-------------|-----------|--------|-------------|------------|
|                | - | - | - | FVLD<br>FJn | CMF<br>Jn | OVFJn  | FDOV<br>FJn | FDF<br>FJn |
| 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: n = 0. 1

Only 0 can be written to this bit after it is read as 1 to clear it.
 Writing 1 is ignored.

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 5 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial<br>Value | R/W       | Description  |
|-----|----------|------------------|-----------|--|
| 4   | FVLDFJn  | 0                | R         | FIFO Status Flag Jn  |
|     |          |                  |           | Indicates whether the FIFO is enabled or disabled. This bit cannot be set or cleared by software.  |
|     |          |                  |           | When this bit is 1, the corresponding FIFO Jn (FIFOJn) is enabled. The value in TCNTJn is latched in the FIFO on the edge of the TIJn pin.   |
|     |          |                  |           | When FIFOENJn = 1, this bit is always set to 1. When FIFOENJn = 0 and FVCRENJn = 1, FVLDFJn becomes 1 on OCRJn compare match. When the FIFO becomes full, FVLDFJn is cleared to 0. When FIFOENJn = 0 and FVCRENJn = 0, this bit is always 0. |
|     |          |                  |           | 0: FIFO is not enabled   |
|     |          |                  |           | [Clearing conditions]  |
|     |          |                  |           | When FIFOENJn is cleared to 0  |
|     |          |                  |           | <ul> <li>When FIFOENJn is 0, FVCRENJn is 1, and the number<br/>of data words in FIFO exceeds the value set in<br/>FDFTRGJn</li> </ul>  |
|     |          |                  |           | 1: FIFO is enabled   |
|     |          |                  |           | [Setting conditions]   |
|     |          |                  |           | When FIFOENJn is set to 1  |
|     |          |                  |           | When FIFOENJn is 0, FVCRENJn is 1, and OCRJn   |
|     | 01451    |                  | D (0.40 ) | compare match occurred   |
| 3   | CMFJn    | 0                | R/(W)*    | Compare Match Flag Jn  |
|     |          |                  |           | Indicates that compare match between TCNTJn and OCRJn has occurred. When this bit is 1, compare match on compare match register Jn (OCRJn) has occured. This bit cannot be set to 1 by software.   |
|     |          |                  |           | Compare match between OCRJn and TCNTJn has not occurred  |
|     |          |                  |           | [Clearing condition]   |
|     |          |                  |           | When writing 0 to this bit after reading it as 1   |
|     |          |                  |           | Compare match between OCRJn and TCNTJn has occurred  |
|     |          |                  |           | [Setting condition]  |
|     |          |                  |           | When the values in TCNTJn and OCRJn match  |

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |
|-----|----------|------------------|--------|---|
| 2   | OVFJn    | 0                | R/(W)* | Overflow Flag Jn  |
|     |          |                  |        | Indicates that timer counter Jn (TCNTJn) has overflowed.  |
|     |          |                  |        | When this bit is 1, timer counter Jn (TCNTJn) has overflowed. This bit cannot be set to 1 by software.  |
|     |          |                  |        | 0: TCNTJn has not overflowed  |
|     |          |                  |        | [Clearing condition]  |
|     |          |                  |        | When writing 0 to this bit after reading it as 1  |
|     |          |                  |        | 1: TCNTJn has overflowed  |
|     |          |                  |        | [Setting condition]   |
|     |          |                  |        | <ul> <li>When TCNTJn has overflowed (from H'FFFF to<br/>H'0000)</li> </ul>  |
| 1   | FDOVFJn  | 0                | R/(W)* | FIFO Data OVerflow Flag Jn  |
|     |          |                  |        | Indicates that another capture has occurred while nine words of data was in the FIFO. When this bit is 1, FIFOJn has overflowed and captured data is lost. This bit cannot be set to 1 by software. |
|     |          |                  |        | 0: Data in the FIFO is normal   |
|     |          |                  |        | [Clearing condition]  |
|     |          |                  |        | • When writing 0 to this bit after reading it as 1  |
|     |          |                  |        | Another capture has occurred while nine words of data was in the FIFO   |
|     |          |                  |        | [Setting condition]   |
|     |          |                  |        | When another capture has occurred while nine words<br>of data was in the FIFO   |

| Bit | Bit Name | Initial<br>Value | R/W    | Description  |
|-----|----------|------------------|--------|--|
| 0   | FDFFJn   | 0                | R/(W)* | FIFO Data Full Flag Jn   |
|     |          |                  |        | Indicates that the number of data words (TCNTJn counter values) captured in the FIFO is equal to or exceeds the value set in the FDFTRGJn bits in the FIFO control register Jn (FCRJn).  |
|     |          |                  |        | When this bit is 1, the number of data words captured in<br>the FIFO is equal to or exceeds the set value and the<br>captured data is ready to be read.  |
|     |          |                  |        | This bit cannot be set to 1 by software.   |
|     |          |                  |        | To clear the flag, write 0 to this bit after reading it as 1. However, the data in FIFOJn must be read to make the number of data words in the FIFO less than the value set in the FDFTRGJn bits. Writing 0 before reading it as 1 is ignored.   |
|     |          |                  |        | When data is read by DMA transfer, this bit is cleared to 0. The transfer request is issued by the FIFO data full interrupt. However, if the number of data words in the FIFO is still equal to or exceeds the value set in the FDFTRGJn bits after the reading by DMA transfer, this bit is set to 1 again. |
|     |          |                  |        | 0: Number of data words in FIFO is less than the value in FDFTRGJn   |
|     |          |                  |        | [Clearing conditions]  |
|     |          |                  |        | • When writing 0 to this bit after reading it as 1   |
|     |          |                  |        | <ul> <li>When DMAC receives DMA transfer request by FIFO data full interrupt</li> </ul>  |
|     |          |                  |        | Number of data words in FIFO is equal to or exceeds the value in FDFTRGJn  |
|     |          |                  |        | [Setting condition]  |
|     |          |                  |        | When the number of data words in FIFO has reached<br>the value in FDFTRGJn   |

Notes: n = 0, 1

\* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

# 13.34.5 Timer Interrupt Enable Registers J0 and J1 (TIERJ0 and TIERJ1)

TIERJ0 and TIERJ1 are 8-bit readable/writable registers that enable and disable interrupt requests for the status flag in timer status register J (TSRJ). Data in the FIFO can be read with DMA transfer requested by the FIFO data full interrupt.

TIERJ0 and TIERJ1 can be read from and written to in byte or word units.

TIERJ0 and TIERJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2         | 1           | 0          |
|----------------|---|---|---|---|---|-----------|-------------|------------|
|                | 1 | - | - | 1 | 1 | OV<br>EJn | FDOV<br>EJn | FDF<br>EJn |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0         | 0           | 0          |
| R/W:           | R | R | R | R | R | R/W       | R/W         | R/W        |

Note: n = 0, 1

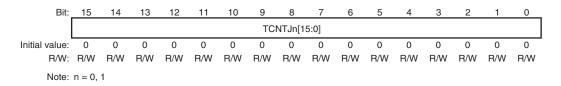
|          | Initial        |   |   |  |  |
|----------|----------------|---|---|--|--|
| Bit Name | Value          | R/W   | Description   |  |  |
| _        | All 0          | R   | Reserved  |  |  |
|          |                |   | These bits are always read as 0. The write value should always be 0.  |  |  |
| OVEJn    | 0              | R/W   | Overflwo Interrupt Enable Jn  |  |  |
|          |                |   | Enables and disables interrupt requests for the status flag (OVEJn) of overflow on timer counter Jn (TCNTJn).   |  |  |
|          |                |   | 0: OVEJn interrupt request is disabled  |  |  |
|          |                |   | 1: OVEJn interrupt request is rnabled   |  |  |
| FDOVEJn  | 0              | R/W   | FIFO Data Overflow Interrupt Enable Jn  |  |  |
|          |                |   | Enables and disables interrupt requests for the status flag (FDOVEJn) of overflow on the FIFO register Jn (FIFOJn)  |  |  |
|          |                |   | 0: FDOVEJn interrupt request is disabled  |  |  |
|          |                |   | 1: FDOVEJn interrupt request is enabled   |  |  |
| FDFEJn   | 0              | R/W   | FIFO Data Full Interrupt Enable Jn  |  |  |
|          |                |   | Enables and disables interrupt requests for the status flag (FDFEJn) of data full of the FIFO register Jn (FIFOJn). While the interrupt request is enabled, set the DMAC. DMA transfer can be activated by the interrupt request. |  |  |
|          |                |   | 0: FDFEJn interrupt request is disabled   |  |  |
|          |                |   | 1: FDFEJn interrupt request is enabled  |  |  |
|          | OVEJn  FDOVEJn | Bit Name Value  — All 0  OVEJn 0  FDOVEJn 0 | Bit Name     Value     R/W       —     All 0     R       OVEJn     0     R/W       FDOVEJn     0     R/W  |  |  |

#### 13.34.6 Timer Counter J0 and J1 (TCNTJ0 and TCNTJ1)

TCNTJ0 and TCNTJ1 are 16-bit readable/writable reigsters. These registers are provided one for each subblock and are incremented by the clook selected from lines 0 to 5 of the clock bus depending on the control register. The counter value is cleared to H'0000 by the edge of pin TIJn.

TCNTJ0 and TCNTJ1 can be read from and wirtten to in word units.

TCNTJ0 and TCNTJ1 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.



| Bit     | Bit Name | Initial<br>Value | R/W | Description                            |
|---------|----------|------------------|-----|--|
| 15 to 0 | TCNTJn   | All 0            | R/W | Timer Counter Jn                       |
|         | [15:0]   |                  |     | These bits store the up-counter value. |

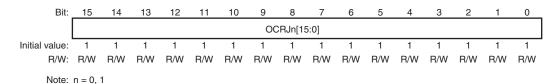
#### 13.34.7 Compare Match Registers J0 and J1 (OCRJ0 and OCRJ1)

OCRJ0 and OCRJ1 are 16-bit readable/writable reigsters. These registers are provided one for each subblock and function as output compare registers for timer counter Jn (TCNTJn).

When FIFOENJn and FVCRENJn in FIFO control register Jn (FCRJn) is 0 and 1, respectively. The FIFO is enabled (FVLDFJn is set to 1) on OCRJn compare match.

OCRJ0 and OCRJ1 can be read from and wirtten to in word units.

OCRJ0 and OCRJ11 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.



| Bit     | Bit Name | Initial<br>Value | R/W   | Description                               |
|---------|----------|------------------|-------|---|
|         | Dit Hamo | raido            | 14,11 | Becomption                                |
| 15 to 0 | OCRJn    | All 1            | R/W   | Compare Match Jn                          |
|         | [15:0]   |                  |       | These bits store the compare match value. |

### 13.34.8 FIFO Registers J0 and J1 (FIFOJ0 and FIFOJ1)

FIFOJ0 and FIFOJ1 are 16-bit read-only registers. These register are provided one for each subblock and can store nime words of data of timer counter Jn (TCNTJn). FIFOJn can be read but cannot be written by the CPU. If FIFOJn in which no data is captured is read, the read value is undefined.

If another capture occurs while nine words of data is cpatured in the FIFO, the latest data is lost. Secure the free area by reading the FIFO before another capture has occured.

FIFOJ0 and FIFOJ1 can be read in word units.

FIFOJ0 and FIFOJ1 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

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

Note: n = 0, 1

| Bit     | Bit Name | Initial<br>Value | R/W | Description                              |
|---------|----------|------------------|-----|--|
| 15 to 0 | FIFODJn  | All 0            | R   | FIFO Data Jn                             |
|         | [15:0]   |                  |     | These bits store the FIFO register data. |

# 13.34.9 FIFO Data Count Registers J0 and J1 (FDNRJ0 and FDNRJ1)

FDNRJ0 and FDNRJ1 are 8-bit read-only registers. These registers are provided one for each subblock and indicate the number of data words in FIFOJ which stores captured data. This register cannot be written to.

This register is incremented by 1 every capture in FIFO and is decremented by 1 every read from FIFO (one-word read).

When the FIFO is reset by the FRSTJn bit in FIFO control register Jn (FCRJn), the value in this register is also cleared to H'0.

FDNRJ0 and FDNRJ1 can be read from in byte units.

FDNRJ0 and FDNRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2    | 1      | 0 |
|----------------|---|---|---|---|---|------|--------|---|
|                | - | - | - | - |   | FDNJ | n[3:0] |   |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0    | 0      | 0 |
| R/W:           | R | R | R | R | R | R    | R      | R |

Note: n = 0, 1

| Bit    | Bit Name   | Initial<br>Value | R/W | Description  |
|--------|------------|------------------|-----|--|
| 7 to 4 | _          | All 0            | R   | Reserved   |
|        |            |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 3 to 0 | FDNJn[3:0] | 0000             | R   | FIFO Data Count Jn   |
|        |            |                  |     | These bits indicate the number of data words captured in the FIFO.   |
|        |            |                  |     | The value ranges from H'0 to H'9. H'0 indicates no data is in the FIFO and H'9 indicates that FIFOJn is full with the captured data. |

### 13.34.10 Noise Canceler Counters J0 and J1 (NCNTJ0 and NCNTJ1)

NCNTJ0 and NCNTJ1 are 8-bit readable/writable registers.

These registers are incremented by an assertion of the input signals (TIJ0 and TIJ1) as triggers when the noise canceler function is enabled by the noise canceler enable bits (NCEJ1 and NCEJ0) in timer control register J (TCRJn). These registers are driven by the noise canceler counter clock, which is supplied by the prescaler.

Premature-transition cancellation or minumum time-at-level cancellation is performed depending on the setting in the noise cancellation mode register (NCMR) of the common controller.

• In premature-transition cancellation mode

When the NCEJn bit is 1, NCNTJn is stopped, and a level change on pin TIJn is detected, NCNTJn is started for counting up. The counter is cleared to H'00 and stopped on the first edge of the  $P\varphi$  clock after the counter value maches the value in noise canceler register Jn (NCRJn).

NCNTJn is incremented regardless of the TJE bit in the ATU-III master enable register (ATUENR).

The level change at the start of counting is output as the signal after noise removal and its edge is to be extracted. However, since subsequent level changes are masked, the noise removal signal is not changed. When the values in the counter and NCRJn match, the input signal level at this time is output as the noise removal signal.

When NCEJn bits are cleared to 0 while the counter is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

• In minumum time-at-level cancellation mode

When the NCEJn bit is 1, NCNTJn is stopped, and a level change on pin TIJn is detected, NCNTJn is started for counting up. If subsequent level change is detected or the values in the counter and the noise canceler register Jn (NCRJn) match, the counter is cleared to H'00 and stopped on the next  $P\phi$  clock cycle.

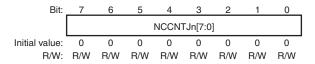
NCNTJn is incremented regardless of the TJE bit in the ATU-III master enable register (ATUENR).

The signal after noise removal is changed only on compare match between the counter and NCRJn in synchronization with the level change at the start of counting. When the counter is stopped before the compare match, level changes at the start and ene of counting are masked and the signal after noise removal is not changed.

When the NCEJn bit is cleared during the counter in operation, counting continues until compare match or a level change on the pin is detected.

NCNTJ0, 1 can be read from and written to in byte or word units.

NCNTJ0, 1 is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Note: n = 0, 1

| Bit    | Bit Name | Initial<br>Value | R/W | Description                               |
|--------|----------|------------------|-----|---|
| 7 to 0 | NCCNTJn  | All 0            | R/W | Noise Canceler Count Jn                   |
|        | [7:0]    |                  |     | These bits store the 8-bit counter value. |

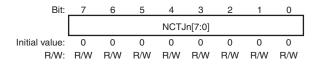
### 13.34.11 Noise Cancel Registers J0 and J1 (NCRJ0 and NCRJ1)

NCRJ0 and NCRJ1 are 8-bit readable/writable registers that set the upper limitation of the noise canceler couters (NCNTJ1 and NCNTJ0). For example, when H'FF is set in these register, a pulse width of up to 1.64 ms is treated as noise ( $P\phi = 20 \text{ MHz}$ ).

Premature-transition cancellation or minumum time-at-level cancellation is performed depending on the setting in the noise cancellation mode register (NCMR) of the common controller. For details, see section 13.34.10, Noise Canceler Counters J0 and J1 (NCNTJ0 and NCNTJ1).

NCRJ0 and NCRJ1 can be read from and written to in byte or word units.

NCRJ0 and NCRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.



Note: n = 0.1

| Bit    | Bit Name   | Initial<br>Value | R/W | Description  |
|--------|------------|------------------|-----|--|
| 7 to 0 | NCTJn[7:0] | All 0            | R/W | Noise Cancellation Time Jn                               |
|        |            |                  |     | These bits store the 8-bit TIJn noise cancellation time. |

# 13.35 Operations of Timer J

The TCNTJn counter clock is selected by the TCRJn register from the clock-bus lines. When the TJE bit in ATUENR and the STRJn bit in TSTR is set to 1, TCNTJn can be operated. TCNTJn is cleared to H'00 by the input edge of pin TIJn. For the edge detection, the rising, falling, or both edge sensing can be selected by the IOJn bit in TCRJn.

When the FIFOEN bit in FCRFJn is set to 1, FIFOJn captures the TCNTJn counter value by the input edge of pin TIJn. The number of data captured in the FIFO is indicated in FDNRn.

When the number of data captured in FIFOJn exceeds the value in the FDFTRGJn bits in FCRJn, the FDFFJn bit in TSRJn is set to 1.

When FIFOENJn = 0, FVCRENJn = 1, and compare match between TCNTJn and OCRJn occurs, FIFOJn is enabled (FVLDFJn in TSRJn is 1). Under this conditions, when the number of data captured in FIFOJn exceeds the value in the FDFTRGJn bits in FCRJn, FIFOJn is disabled.

Operation examples for subblock 0 are shown below. The TCNTJ0 counter clock is an ideal signal to show the counting and clearing timing of TCNTJ0.

In figure 13.50, falling edge sensing is selected, FIFOEN = 1, FDFTRGJ0 = B'11, and FDFFJ0 is set by the input edge.

In figure 13.51, falling edge sensing is selected, FIFOEN = 0, FVCRENJ0 = 1, FDFTRGJ0 = B'00 and FDFF0 is set by the input edge.

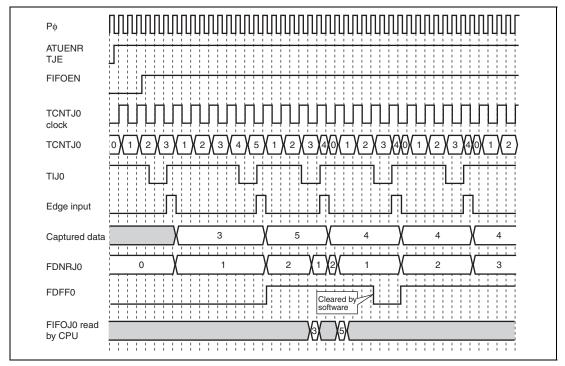


Figure 13.50 Operation Example of Timer J (1)

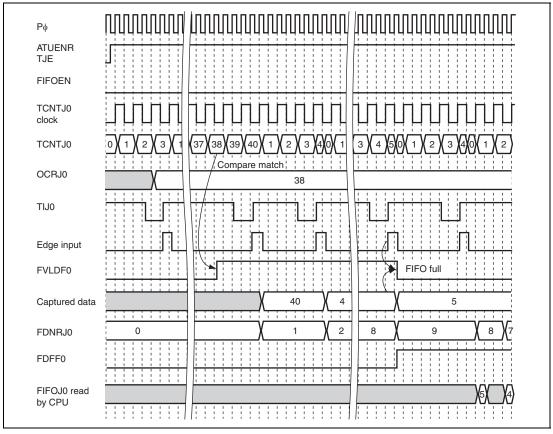


Figure 13.51 Operation Example of Timer J (2)

# 13.36 Usage Notes

Note that the kinds of conflicts and operation described below occur during ATU-III operation.

In this section, n denotes the subblock number and m denotes the channel number, for each timer. The values of n and m differ for each timer. For details, refer to the description of each timer.

### 13.36.1 Input Capture Conflict Operation

### (1) Conflict between Writing to General Register and Input Capture

When a write to a general register occurs simultaneously with input capture, writing takes priority (waveforms in the left half of figure 13.52). However, if the input capture status is provided, the input capture status flag is set.

The waveforms in the right half of figure 13.52 indicate a case in which writing occurs one  $P\phi$  cycle prior to input capture.

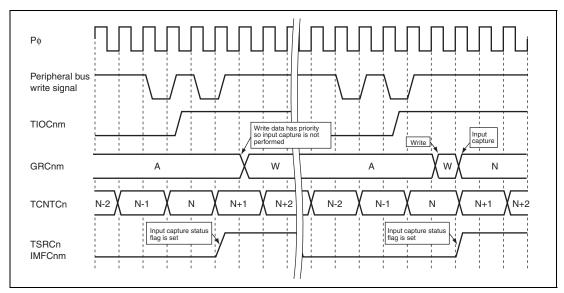


Figure 13.52 Conflict between Writing to GRCnm and Input Capture

Table 13.18 Resources Related to Conflict Operation between General Register Write and Input Capture

| Timer   | Counter<br>(Whose Value is Captured) | Capture Register | Status      |
|---------|--------------------------------------|------------------|-------------|
| Timer C | TCNTCn                               | GRCnm            | IMFCnm      |
| Timer D | TCNT2Dn                              | GRDnm            | CMFBDnm     |
| Timer F | ECNTAFn                              | GRAFn            | ICFFn       |
|         | ECNTBFn                              | GRBFn            |             |
|         | ECNTCFn                              | GRCFn            |             |
|         | ECNTCFn + GRDFn                      | GRDFn            | <del></del> |

### (2) Conflict between Writing to Counter and Input Capture

When a write to a counter occurs simultaneously with input capture, the value immediately before writing is captured (waveforms in the left half of figure 13.53). The waveforms in the right half of figure 13.53 indicate a case in which writing occurs one  $P\phi$  cycle prior to input capture, so the written value is captured.

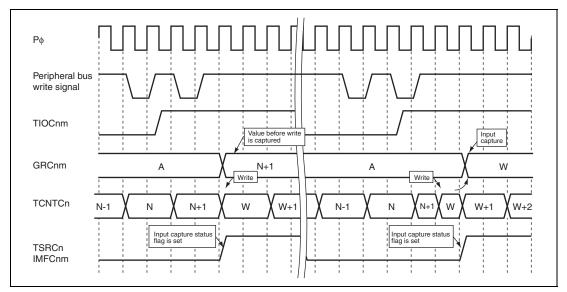


Figure 13.53 Conflict between Writing to TCNTCn and Input Capture

Table 13.19 Resources Related to Conflict Operation between Counter Write and Input Capture

| Timer   | Counter<br>(Whose Value is Captured) | Capture Register | Status             |
|---------|--------------------------------------|------------------|--------------------|
| Timer A | TCNTAn                               | ICRAn            | ICFAn              |
| Timer B | TCNTB0                               | ICRB0            | ICFB0              |
|         | TCNTB0 + ICRB1                       | ICRB1            | <del></del>        |
| Timer C | TCNTCn                               | GRCnm            | IMFCnm             |
| Timer D | TCNT1Dn                              | OSBRDn           | _                  |
|         | TCNT2Dn                              | GRDnm            | _                  |
| Timer F | ECNTAFn                              | GRAFn            | ICFFn              |
|         | ECNTBFn                              | GRBFn            | <del></del>        |
|         | ECNTCFn                              | GRCFn            | <del></del>        |
|         | ECNTCFn + GRDFn                      | GRDFn            | <del></del>        |
| Timer J | TCNTJn                               | FIFOJn           | FDOVFJn,<br>FDFFJn |

### (3) Conflict between Setting and Clearing of Input Capture Status Flag

Flag clearing by writing 0 to it after reading it as 1 or by using the ACK signal from the DMAC takes priority over flag setting by input capture. The waveforms in the left half of figure 13.54 indicate an example in which input capture occurs simultaneously with writing 0 to the status flag to clear it, and the status flag is cleared as a result. In contrast with this, the waveforms in the right half of figure 13.54 indicate an example in which input capture is performed immediately after the flag has been cleared.

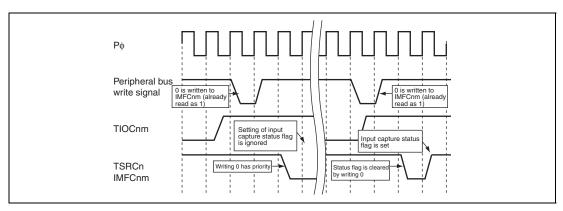


Figure 13.54 Conflict between Status Flag Clearing by Writing 0 and Input Capture

Table 13.20 Resources Related to Conflict Operation between Status Flag Clearing by Writing 0 and Input Capture

| Timer   | Flag            | Timer   | Flag  |
|---------|-----------------|---------|-------|
| Timer A | ICFAn           | Timer B | ICFB0 |
| Timer C | IMFCnm          | Timer F | ICFFn |
| Timer J | FDOVFJn, FDFFJn | _       | _     |

Figure 13.55 shows an example in which input capture conflicts with clearing of the status flag by the ACK signal from the DMAC.

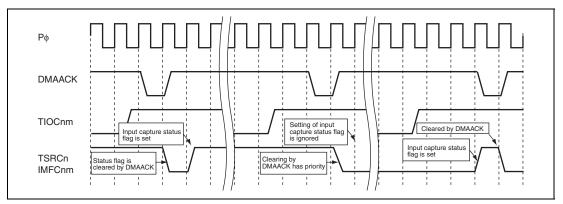


Figure 13.55 Conflict between Status Flag Clearing by DMAACK and Input Capture

Table 13.21 Resources Related to Conflict Operation between Status Flag Clearing by DMAACK and Input Capture

| Timer   | Flag   |
|---------|--------|
| Timer C | IMFCn0 |
| Timer J | FDFFJn |

# 13.36.2 Compare Match Conflict Operation

### (1) Conflict between Writing to Compare-Match General Register and Compare Match

A conflict between a write to a register provided with the compare match function and compare match is described here with timer C used as an example.

If writing is performed after GRCnm and TCNTCn have matched (waveforms in the left half of figure 13.56), the compare match status flag is set. If GRCnm and TCNTCn do not match for even one  $P\phi$  cycle (waveforms in the right half of figure 13.56), no compare match is detected.

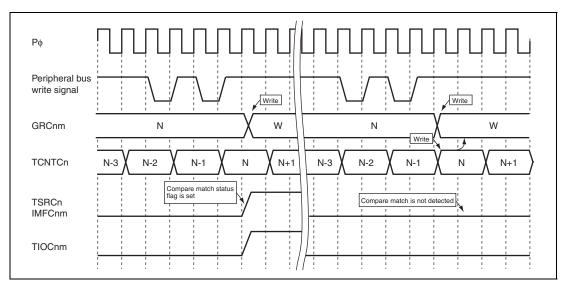


Figure 13.56 Conflict between Writing to GRCnm and Compare Match

Table 13.22 Resources Related to Conflict Operation between Write to Compare-Match General Register and Compare Match

| Timer   | Counter | Compare-Match Register | Status  |
|---------|---------|------------------------|---------|
| Timer B | TCNTB1  | OCRB1                  | CMFB1   |
| Timer C | TCNTCn  | GRCnm                  | IMFCnm  |
| Timer D | TCNT1Dn | OCRDnm                 | CMFADnm |
|         | TCNT2Dn | GRDnm                  | CMFBDnm |
| Timer F | ECNTAFn | GRAFn                  | _       |
|         | ECNTBFn | GRBFn                  | _       |
|         | ECNTCFn | GRBFn                  | OVFCFn  |
| Timer G | TCNTGn  | OCRGn                  | CMFGn   |
| Timer H | TCNT1H  | OCR1H                  | CMFH    |
| Timer J | TCNTJn  | OCRJn                  | CMFJn   |

Note: Timing of compare match B0 and compare match B6 of timer B and the cycle match of timer E differ from the timing of these compare matches. For details, see section 13.36.2 (2), Conflict between CYLREnm Write and Cycle Match of CYLREnm—TCNTEnm.

### (2) Conflict between CYLREnm Write and Cycle Match of CYLREnm-TCNTEnm

Operation when a write to CYLREnm occurs simultaneously with compare match (cycle match) with TCNTEnm is shown below. As the waveforms in the left half of figure 13.57 indicate, if CYLREnm is written to at the same time the counter is cleared by cycle match, TCNTEnm is cleared as is done on normal cycle match, and the cycle match status and PWM output also change. The waveforms in the right half of figure 13.57 show an example in which CYLREnm is written to before the counter is cleared. In this case, no cycle match is detected and TCNTEnm continues to be incremented.

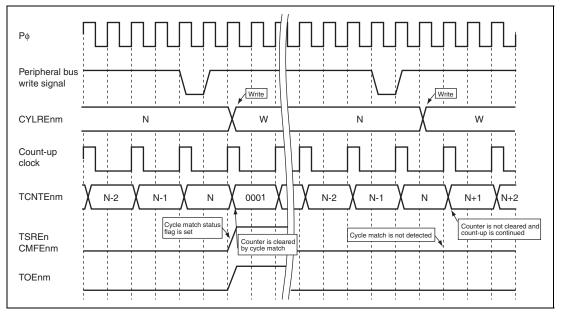


Figure 13.57 Conflict between Writing to CYLREnm and Cycle Match

Table 13.23 Resources Related to Conflict Operation between Writing to Cycle Setting Register and Cycle Match with Timer Counter

| Timer   | Counter | Compare (Cycle) Match Register | Status |
|---------|---------|--------------------------------|--------|
| Timer B | TCNTB0  | OCRB0                          | CMFB0  |
|         | TCNTB6  | OCRB6/OCRB7                    | CMFB6  |
| Timer E | TCNTEnm | CYLREnm                        | CMFEnm |

# (3) Conflict between Writing to Counter and Compare Match

A conflict between a write to a counter and compare match is described below. If writing is performed after the compare match register and counter values have matched (waveforms in the left half of figure 13.58), the compare match status flag is set. If the compare match register and counter do not match for even one  $P\phi$  cycle (waveforms in the right half of figure 13.58), no compare match is detected.

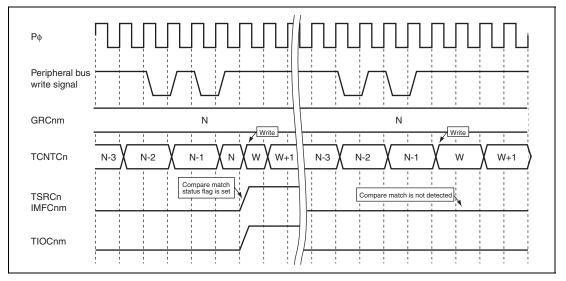


Figure 13.58 Conflict between Writing to TCNTCn and Compare Match

Table 13.24 Resources Related to Conflict Operation between Writing to Counter and Compare Match

| Timer   | Counter | Compare Match Register | Status  |
|---------|---------|------------------------|---------|
| Timer B | TCNTB1  | OCRB1                  | CMFB1   |
| Timer C | TCNTCn  | GRCnm                  | IMFCnm  |
| Timer D | TCNT1Dn | OCRDnm                 | CMFADnm |
|         | TCNT2Dn | GRDnm                  | CMFBDnm |
| Timer F | ECNTAFn | GRAFn                  | _       |
|         | ECNTBFn | GRBFn                  |         |
|         | ECNTCFn | GRBFn                  | OVFCFn  |
| Timer G | TCNTGn  | OCRGn                  | CMFGn   |
| Timer H | TCNT1H  | OCR1H                  | CMFH    |
| Timer J | TCNTJn  | OCRJn                  | CMFJn   |

Note: Timing of compare match B0 and compare match B6 of timer B and the cycle match of timer E differ from the timing of these compare matches. For details, see section 13.36.2 (5), Conflict between Writing to TCNTEnm and Counter Clearing by Cycle Match.

### (4) Conflict between Writing to Counter and Counter Clearing by Compare Match

The waveforms shown here are for when the function to clear a counter by compare match is enabled. When a write to a counter occurs simultaneously with counter clearing by compare match, the counter is not cleared and writing takes priority (waveforms in the left half of figure 13.59). However, the compare match status flag is set. The waveforms in the right half of figure 13.59 show a case in which writing to TCNTCn is one  $P\phi$  cycle later.

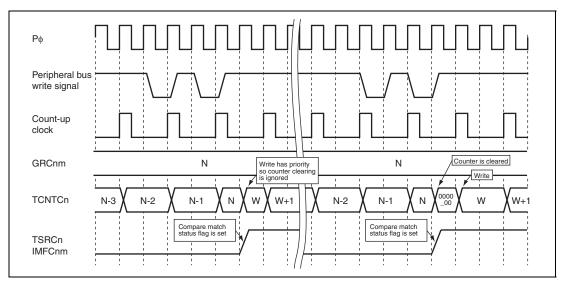


Figure 13.59 Conflict between Writing to TCNTCn and Counter Clearing by Compare Match

Table 13.25 Resources Related to Conflict Operation between Writing to Counter and Counter Clearing by Compare Match

| Timer   | Counter | Compare Match Register | Status |
|---------|---------|------------------------|--------|
| Timer C | TCNTCn  | GRCnm                  | IMFCnm |
| Timer F | ECNTAFn | GRAFn                  | _      |
|         | ECNTBFn | GRBFn                  |        |
| Timer G | TCNTGn  | OCRGn                  | CMFGn  |
| Timer H | TCNT1H  | OCR1H                  | CMFH   |

### (5) Conflict between Writing to TCNTEnm and Counter Clearing by Cycle Match

When a write to TCNTEnm occurs simultaneously with counter clearing by cycle match, the counter is not cleared and TCNTEnm is written to. Note that the cycle match status flag is set, and the cycle setting register and duty cycle setting register are reloaded (waveforms in the right half of figure 13.60). In addition, the PWM waveforms are output as they are in cycle match.

The waveforms in the left half of figure 13.60 show operation when writing is performed one P $\phi$  cycle earlier than the count-up clock.

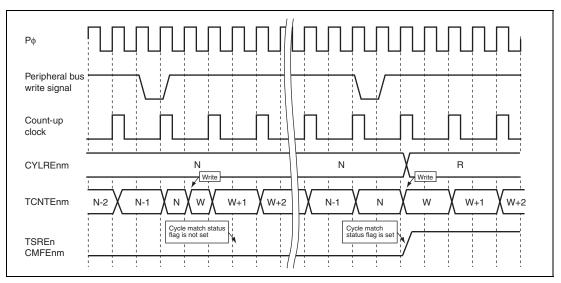


Figure 13.60 Conflict between TCNTEnm Write and Counter Clearing by Cycle Match

Table 13.26 Resources Related to Conflict Operation between Counter Write and Counter Clearing by Cycle Match

| Timer   | Counter | Compare (Cycle) Match Register | Status |
|---------|---------|--------------------------------|--------|
| Timer B | TCNTB0  | OCRB0                          | CMFB0  |
|         | TCNTB6  | OCRB6/OCRB7                    | CMFB6  |
| Timer E | TCNTEnm | CYLREnm                        | CMFEnm |

### (6) Conflict between Setting and Clearing of Compare Match Status Flag

Flag clearing by writing 0 to the flag after reading it as 1 takes priority over flag setting by compare match (waveforms in the left half of figure 13.61). The waveforms in the right half of figure 13.61 show the way compare match occurs immediately after the status flag has been cleared by writing 0 to it.

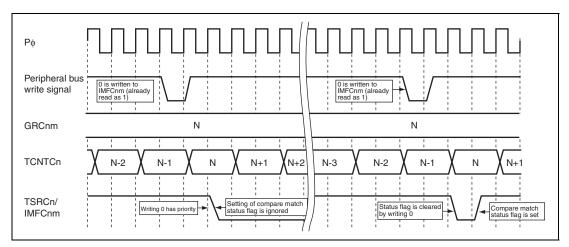


Figure 13.61 Conflict between Setting and Clearing of Compare Match Status Flag

Table 13.27 Resources Related to Conflict Operation between Setting and Clearing of Compare Match Status Flag

| Timer   | Counter | Compare Match Register | Status  |
|---------|---------|------------------------|---------|
| Timer B | TCNTB1  | OCRB1                  | CMFB1   |
| Timer C | TCNTCn  | GRCnm                  | IMFCnm  |
| Timer D | TCNT1Dn | OCRDnm                 | CMFADnm |
|         | TCNT2Dn | GRDnm                  | CMFBDnm |
| Timer G | TCNTGn  | OCRGn                  | CMFGn   |
| Timer H | TCNT1H  | OCR1H                  | CMFH    |
| Timer J | TCNTJn  | OCRJn                  | CMFJn   |

Note: Timing of compare match B0 and compare match B6 of timer B and the cycle match of timer E differ from the timing of these compare matches. For details, see section 13.36.2 (8), Conflict between Setting of Cycle Match Status Flag and Clearing by Writing 0.

### (7) Conflict between Setting of Compare Match Status Flag and Clearing by DMAACK

When setting of the compare match status flag occurs simultaneously with the DMAACK signal, clearing of the status flag due to the DMAACK signal takes priority.

Table 13.28 Resource Related to Conflict Operation between Setting of Compare Match Status Flag and Clearing by DMAACK

| Timer   | Counter | Compare Match Register | Status |
|---------|---------|------------------------|--------|
| Timer G | TCNTGn  | OCRGn                  | CMFGn  |

### (8) Conflict between Setting of Cycle Match Status Flag and Clearing by Writing 0

When setting of the cycle match status flag (cycle match) occurs simultaneously with writing 0 to the status flag after reading it as 1, writing 0 takes priority. The waveforms in the left half of figure 13.62 indicate an example in which setting of the flag due to cycle match occurs simultaneously with clearing of the flag by 0 written to it. The waveforms in the right half of figure 13.62 show an example in which the flag is cleared one  $P\phi$  cycle earlier than flag setting.

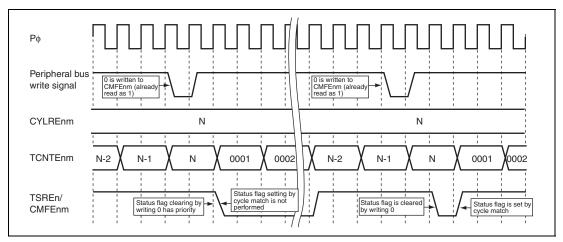


Figure 13.62 Conflict between Clearing of Cycle Match Status Flag by Writing 0 and Cycle Match

Table 13.29 Resources Related to Conflict Operation between Setting of Cycle Match Status Flag and Clearing by Writing 0

| Timer   | Counter | Compare (Cycle) Match Register | Status |
|---------|---------|--------------------------------|--------|
| Timer B | TCNTB0  | OCRB0                          | CMFB0  |
|         | TCNTB6  | OCRB6                          | CMFB6  |
| Timer E | TCNTEnm | CYLREnm                        | CMFEnm |

#### **(9)** Conflict between Setting of Cycle Match Status Flag and Clearing by DMAACK

When setting of the cycle match status flag (cycle match) occurs simultaneously with the DMAACK signal, clearing of the status flag due to the DMAACK signal takes priority.

Table 13.30 Resource Related to Conflict Operation between Setting of Cycle Match Status Flag and Clearing by DMAACK

| Timer   | Counter | Cycle Match Register | Status |
|---------|---------|----------------------|--------|
| Timer E | TCNTEnm | CYLREnm              | CMFEnm |

# (10) Conflict between Status Flag Setting by Forced Compare Match and Status Flag Clearing by DMAACK

When the status flag is set by forced compare match at the same time the status flag is cleared by the DMAACK signal, clearing of the status flag takes priority (waveforms in the left half of figure 13.63). On the other hand, the waveforms in the right half of figure 13.63 show an example in which the DMAACK signal is input to clear the status flag during the cycle in which 1 is written to the forced compare match bit. In this case, the attempt to write 1 to the forced compare match bit is ignored and it is cleared together with the status flag.

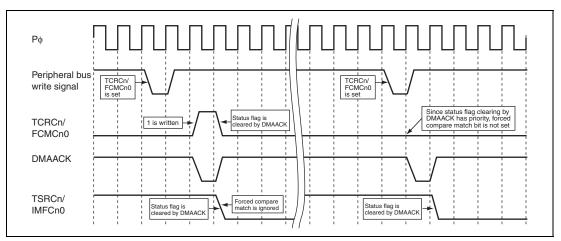


Figure 13.63 Conflict between Forced Compare Match and Status Flag Clearing by DMAACK

Table 13.31 Resource Related to Conflict Operation between Status Flag Setting by Forced Compare Match and Status Flag Clearing by DMAACK

| Timer   | Counter | Compare Match Register | Status |
|---------|---------|------------------------|--------|
| Timer C | TCNTCn  | GRCnm                  | IMFCnm |

# (11) Conflict between Detection of 1H Compare Match and Disabling of Counter by **ATUENR Setting**

When compare match between TCNT1H and OCR1H is detected at the same time a counter enable bit (each bit in ATUENR) is changed to 0, compare match is detected but TCNT2H is not incremented (waveforms in the left half of figure 13.64). Thereafter, even though a counter enable bit is set to 1, TCNT2H is not incremented until the next compare match occurs.

The waveforms in the right half of figure 13.64 show an example in which a counter enable bit is changed to 0 one P\psi cycle after compare match has been detected.

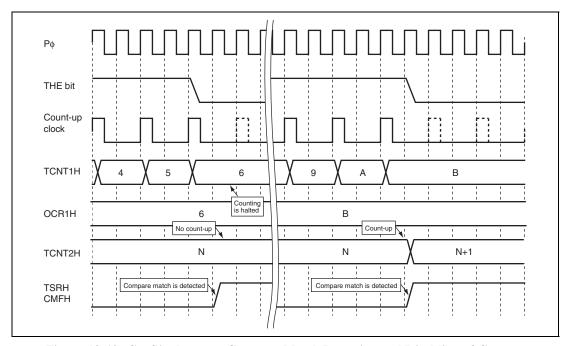


Figure 13.64 Conflict between Compare Match Detection and Disabling of Counter

Table 13.32 Resource Related to Conflict Operation between Compare Match Detection and **Clearing of Counter Enable Bit** 

| Timer   | Counter | Compare Match Register | Status |
|---------|---------|------------------------|--------|
| Timer H | TCNT1H  | OCR1H                  | CMFH   |

### (12) Conflict between Writing 0 to TCNTEnm and Cycle Match

Operation when writing 0 to TCNTEnm occurs simultaneously with cycle match is shown below. The waveforms in the left half of figure 13.65 show a case in which H'0000 is written to TCNTEnm at the same time TCNTEnm is to be cleared to H'0001 due to cycle match. Though the cycle match status flag is set, PWM output is not started because writing 0 takes priority. PWM output is restarted when TCNTEnm is incremented to H'0001.

The waveforms in the middle of figure 13.65 indicate a case in which 0 is written to TCNTEnm one  $P\phi$  cycle after the counter has been cleared by cycle match. Cycle match detection and PWM output are restarted at the timing the TCNTEnm counter value changes from N to 1. In contrast with this, the waveforms in the right half of figure 13.65 show an example in which 0 is written one  $P\phi$  cycle before detection of cycle match. In this case, neither cycle match is detected nor PWM output restarted, and the previous state is retained.

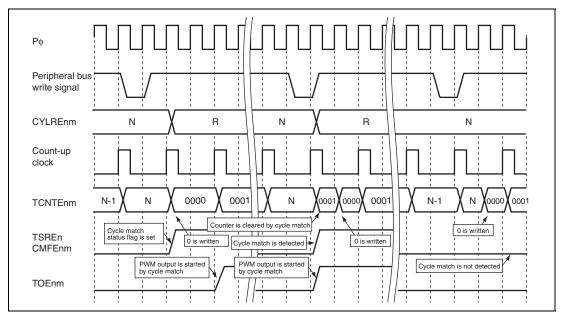


Figure 13.65 Conflict between Writing 0 to TCNTEnm and Cycle Match

### 13.36.3 Load/Reload Conflict Operation

#### **(1)** Conflict between Data Transfer and Writing to Transfer Destination Register

A conflict between data transfer between registers and a peripheral bus write to the transfer destination register is described below.

When data transfer occurs simultaneously with a write to the transfer destination register, writing takes priority and the attempt of data transfer is ignored. Figure 13.66 shows a conflict between reload to CYLREnm of timer E and a write to it.

As shown by the waveforms in the left half of figure 13.66, if writing to CYLREnm occurs at the same timing as cycle reload, writing takes priority. The waveforms in the right half of figure 13.66 indicate a case in which CYLREnm is written to immediately after cycle reload.

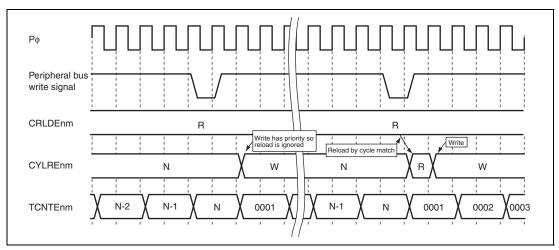


Figure 13.66 Conflict between Writing to CYLREnm and Cycle Reload

Table 13.33 Resources Related to Conflict Operation between Data Transfer and Writing to Transfer Destination Register

| Timer   | Transfer Data | Transfer Destination Register | Transfer Timing |
|---------|---------------|-------------------------------|-----------------|
| Timer B | ICRB0         | TCNTB2                        | External event  |
|         | LDB           |                               |                 |
|         | TCNTB2 - PIMR |                               |                 |
|         | TCNTB2 + RLDB |                               |                 |
|         | ICRB0 - PIMR  | RLDB                          | External event  |
|         | LDB – PIMR    |                               |                 |
|         | TCNTB3 + PIMR | TCNTB3                        | External event  |
|         | TCNTB3        | TCNTB4                        | External event  |
| Timer E | CRLDEnm       | CYLREnm                       | Cycle match     |
|         | DRLDEnm       | DTREnm                        | Cycle match     |

### (2) Conflict between Data Transfer and Writing to Transfer Source Register

A conflict between data transfer between registers and a peripheral bus write to the transfer source register is described below. When data transfer occurs simultaneously with a write to the transfer source register, the value prior to writing is transferred. At the same time, the value of the transfer source register is modified. Operation when writing to CRLDEnm occurs at the timing of cycle reload is shown below. If writing to CRLDEnm occurs at the same timing as cycle reload (waveforms in the left half of figure 13.67), the value immediately before writing is reloaded. On the other hand, the waveforms in the right half of figure 13.67 show an example in which CRLDEnm is written to one cycle earlier than cycle reload.

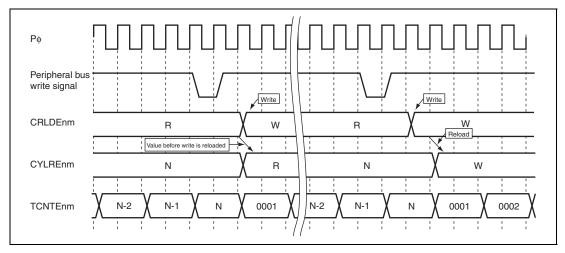


Figure 13.67 Conflict between Writing to CRLDEnm and Cycle Reload

Table 13.34 Resources Related to Conflict Operation between Data Transfer and Writing to Transfer Source Register

| Timer   | Transfer Source<br>Register | Transferred Value | Transfer Destination Register | Transfer Timing |
|---------|-----------------------------|-------------------|-------------------------------|-----------------|
| Timer B | LDB                         | LDB – PIMR        | RLDB                          | External event  |
|         |                             | LDB               | TCNTB2                        | External event  |
|         | PIMR                        | ICRB0 – PIMR      | RLDB                          | External event  |
|         |                             | LDB – PIMR        |                               |                 |
|         |                             | TCNTB2 - PIMR     | TCNTB2                        | External event  |
|         |                             | TCNTB3 + PIMR     | TCNTB3                        | External event  |
|         | RLDB                        | TCNTB2 + RLDB     | TCNTB2                        | External event  |
|         | TCNTB3                      | TCNTB3            | TCNTB4                        | External event  |
| Timer E | CRLDEnm                     | CRLDEnm           | CYLREnm                       | Cycle match     |
|         | DRLDEnm                     | DRLDEnm           | DTREnm                        | Cycle match     |

# 13.36.4 Counter Conflict Operation

### (1) Conflict between Writing to Counter and Count-Up/Count-Down

When a write to a counter occurs simultaneously with incrementation/decrementation of the counter, the write operation takes priority. The attempt to increment/decrement the value is ignored and incrementation/decrementation recommences from the new value on the next counter clock.

### (2) Conflict between Count-Up and Counter Clearing

When incrementation of a counter occurs simultaneously with clearing of the counter, the counter is not cleared to 0 but cleared to 1.

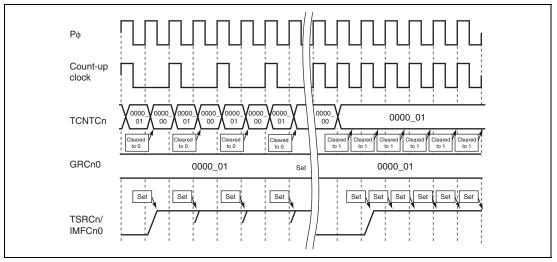


Figure 13.68 Simultaneous Occurrence of Count-Up and Counter Clearing

Table 13.35 Resources Related to Conflict Operation between Count-Up and Counter Clearing

|         |         | Compare Match |                                   |
|---------|---------|---------------|-----------------------------------|
| Timer   | Counter | Register      | Remarks                           |
| Timer C | TCNTCn  | GRCn0         | Only when PWMn0 = 1               |
| Timer F | ECNTAFn | GRAFn         | Only when MDFn = 000, 110, or 111 |
|         | ECNTBFn | GRBFn         | Only when MDFn = 001              |
| Timer G | TCNTGn  | OCRGn         |                                   |
| Timer H | TCNT1H  | OCR1H         |                                   |

#### **(3)** Conflict between Writing to Counter and Overflow

When counter overflow occurs simultaneously with a write to TCNTCn, writing to TCNTCn takes priority. However, the overflow status flag is set (waveforms in the left half of figure 13.69). If the timing for writing to the counter is earlier than incrementation of the counter (waveforms in the right half of figure 13.69), the overflow status flag is not set.

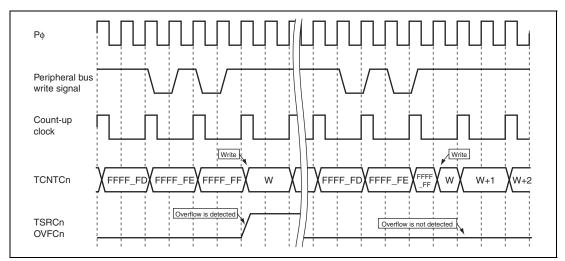


Figure 13.69 Conflict between Writing to TCNTCn and Counter Clearing on Overflow

Table 13.36 Resources Related to Conflict Operation between Writing to Counter and Overflow

| Timer   | Counter | Status |
|---------|---------|--------|
| Timer A | TCNTA   | OVFA   |
| Timer C | TCNTCn  | OVFCn  |
| Timer D | TCNT1Dn | OVF1Dn |
|         | TCNT2Dn | OVF2Dn |
| Timer E | TCNTEnm | OVFEnm |
| Timer F | ECNTAFn | OVFAFn |
|         | ECNTBFn | OVFBFn |
|         | ECNTCFn | OVFCFn |
| Timer G | TCNTGn  | OVFGn  |
| Timer H | TCNT1H  | OVF1H  |
|         | TCNT2H  | OVF2H  |
| Timer J | TCNTJn  | OVFJn  |

### (4) Conflict between Setting and Clearing of Overflow Status Flag

When clearing and setting of the overflow status flag occur simultaneously, clearing takes priority. Shown below is an example in which the status flag is set by the counter value overflowing from H'FFFF FF to H'0000 00 at the same time the status flag is cleared by 0 written to it (waveforms in the left half of figure 13.70). The waveforms in the right half of figure 13.70 show the way the overflow status flag is set again immediately after the status flag has been cleared.

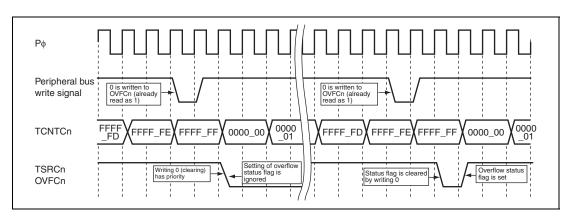


Figure 13.70 Conflict between Setting and Clearing of Overflow Status Flag

Table 13.37 Resources Related to Conflict Operation between Setting and Clearing of Overflow Status Flag

| Timer   | Counter | Status |  |
|---------|---------|--------|--|
| Timer A | TCNTA   | OVFA   |  |
| Timer C | TCNTCn  | OVFCn  |  |
| Timer D | TCNT1Dn | OVF1Dn |  |
|         | TCNT2Dn | OVF2Dn |  |
| Timer E | TCNTEnm | OVFEnm |  |
| Timer F | ECNTAFn | OVFAFn |  |
|         | ECNTBFn | OVFBFn |  |
|         | ECNTCFn | OVFCFn |  |
| Timer G | TCNTGn  | OVFGn  |  |
| Timer H | TCNT1H  | OVF1H  |  |
|         | TCNT2H  | OVF2H  |  |
| Timer J | TCNTJn  | OVFJn  |  |

### (5) Conflict between Overflow and Counter Clearing by Compare Match

If the maximum value is set in a compare match register that has the function to clear a counter by compare match and that function is enabled, when the counter reaches its maximum value, the counter is cleared. No overflow is detected even if the count-up clock frequency is the same as Pφ.

Examples using TCNT1H and OCR1H of timer H are shown in figure 13.71. With H'FFFF set in OCR1H, a case in which the count-up clock is not  $P\phi \times 1/1$  (waveforms in the left half of figure 13.71) and a case in which the count-up clock is  $P\phi \times 1/1$  (waveforms in the right half of figure 13.71) are shown.

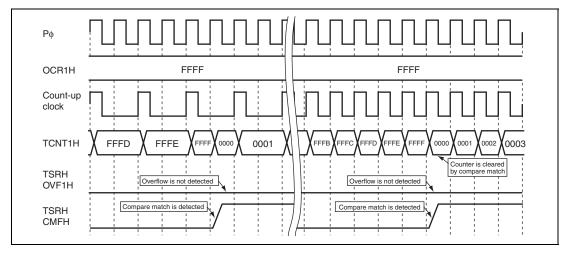


Figure 13.71 Conflict between TCNT1H Counter Overflow and Compare Match

Table 13.38 Resources Related to Conflict Operation between Counter Overflow and Compare Match

| Timer   | Counter | Status | Remarks                           |
|---------|---------|--------|-----------------------------------|
| Timer C | TCNTCn  | OVFCn  | Only when PWMn0 = 1               |
| Timer F | ECNTAFn | OVFAFn | Only when MDFn = 000, 110, or 111 |
|         | ECNTBFn | OVFBFn | Only when MDFn = 001              |
| Timer G | TCNTGn  | OVFGn  |                                   |
| Timer H | TCNT1H  | OVF1H  |                                   |

When the function to clear a counter by compare match is not provided or when that function is disabled, the overflow status flag is set. Figure 13.72 shows operation when the PWMn0 bit of timer C is 1 (counter clearing is enabled) and when the PWMn0 bit is 0 (counter clearing is disabled).

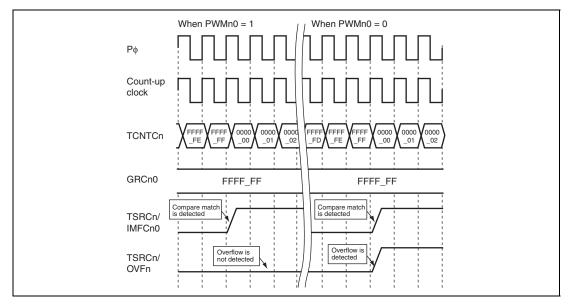


Figure 13.72 Conflict between Counter Clearing by Compare Match of Timer C and Overflow (PWMn0 = 1/0)

### 13.36.5 Noise Canceler Conflict Operation

Conflicts in the noise cancelers are described here.

Table 13.39 Resources Related to Conflict Operation between Writing to Noise Canceler Counter and Compare Match with Noise Canceler Register

| Timer   | Counter        | Compare Match Register |
|---------|----------------|------------------------|
| Timer A | Fimer A NCNTAn |                        |
| Timer C | NCNTCnm        | NCRCnm                 |
| Timer F | NCNTAFn        | NCRAFn                 |
|         | NCNTBFn        | NCRBFn                 |
| Timer J | NCNTJn         | NCRJn                  |

# (1) Conflict between Writing to Noise Canceler Counter and Compare Match with Noise Canceler Register

When a write to NCNT occurs simultaneously with a compare match with NCR, writing takes priority. An example in minimum time-at-level cancellation mode is shown below. In the example in the left half of figure 13.73, since writing prevents compare match from occurring, input capture is also not performed. The example in the right half of figure 13.73 shows a case in which writing is performed one  $P\phi$  cycle later. In this case, compare match occurs so input capture processing is carried out.

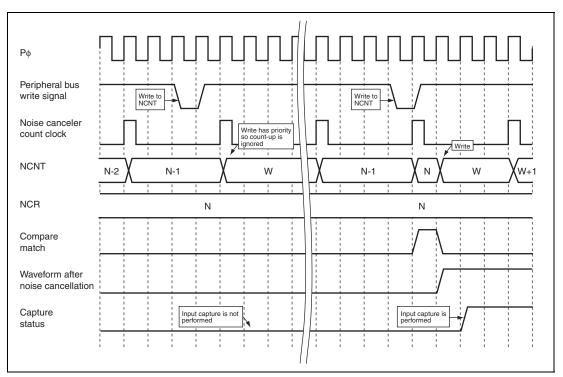


Figure 13.73 Conflict between Writing to NCNT and Compare Match of NCNT-NCR (Example in Minimum Time-At-Level Cancellation Mode)

#### Conflict between Writing to Noise Canceler Register and Compare Match with Noise **(2) Canceler Counter**

When a write to NCR occurs simultaneously with a compare match with NCNT, writing takes priority. An example in minimum time-at-level cancellation mode is shown below. In the example in the left half of figure 13.74, since writing prevents compare match from occurring, input capture is also not performed. The example in the right half of figure 13.74 shows a case in which writing is performed one P $\phi$  cycle later. In this case, compare match occurs so input capture processing is carried out.

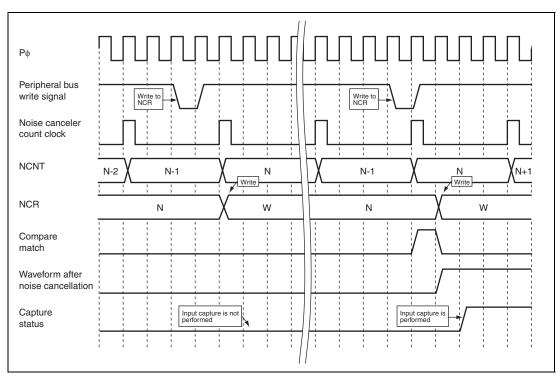


Figure 13.74 Conflict between Writing to NCR and Compare Match of NCR-NCNT (Example in Minimum Time-At-Level Cancellation Mode)

### 13.36.6 Conflict Regarding Down Counter D

Conflicts in DCNTDnm are described here.

### (1) Conflict between Writing to DCNTDnm Counter and Count-Down

When a write to DCNTDnm occurs simultaneously with decrementation of the down counter, writing to DCNTDnm is performed. The attempt to decrement the value is ignored and decrementation recommences from the new value on the next count-down clock.

### (2) Conflict between Writing to DCNTDnm Counter and Underflow

When a write to DCNTDnm occurs simultaneously with underflow, writing to DCNTDnm is performed. The example in the left half of figure 13.75 shows operation when a count-down clock is input simultaneously with a write to DCNTDnm when the DCNTDnm value is H'0000 00. Though the new value is written to DCNTDnm, count-down operation will be halted because underflow is detected. The underflow status flag is set. In the example shown by the waveforms in the right half of figure 13.75, DCNTDnm is written to one  $P\phi$  cycle earlier so underflow is not detected.

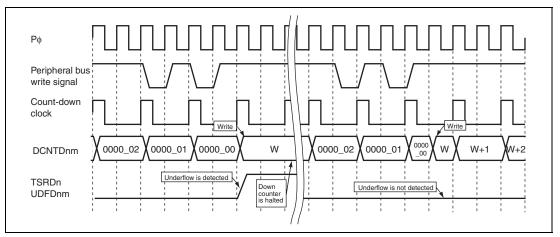


Figure 13.75 Conflict between Writing to DCNTDnm and Underflow

#### Conflict between Writing to DCNTDnm Counter and Compare Match B (Counter Stop **(3)** Trigger)

When a write to DCNTDnm occurs simultaneously with compare match B, writing to DCNTDnm is performed (if compare match B is selected as a condition to stop the down counter).

An example in which writing 0 to DCNTDnm occurs simultaneously with clearing of the counter by detection of compare match B is shown in the middle of figure 13.76. The attempt to clear DCNTDnm by compare match B is ignored and writing takes priority. However, the output on TODnmB is turned off due to compare match B, and DCNTDnm halts with the written value retained.

The waveforms in the right half of figure 13.76 show a case in which the write cycle occurs one P\( \phi\) cycle earlier. During the P\( \phi\) cycle subsequent to writing to DCNTDnm, the counter is cleared by compare match B.

The waveforms in the left half of figure 13.76 show an example in which writing is performed immediately after the counter has been cleared by compare match B.

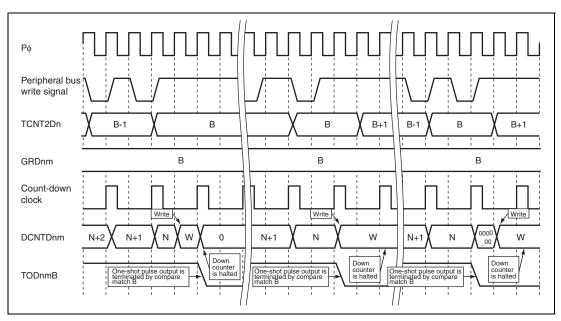


Figure 13.76 Conflict between Writing to DCNTDnm and Counter Clearing by Compare Match B

#### (4) Conflict between Setting of Underflow Status Flag and Clearing by Writing 0

When clearing of the underflow status flag by writing 0 to it occurs simultaneously with underflow, status flag clearing takes priority. Shown below is an example in which setting of the status flag due to underflow of DCNTDnm occurs simultaneously with clearing of the status flag by 0 written to it (waveforms in the left half of figure 13.77). The waveforms in the right half of figure 13.77 show how the status flag is set again by underflow occurrence immediately after the status flag has been cleared.

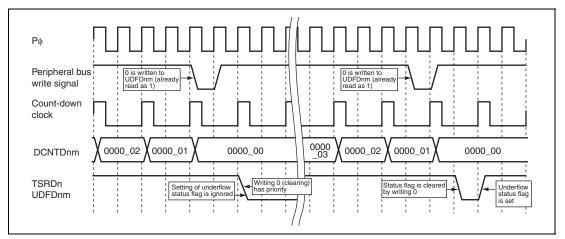


Figure 13.77 Conflict between Setting and Clearing of Underflow Status Flag

### (5) Conflict between Setting of Underflow Status Flag and Clearing by DMAACK

When clearing of the underflow status flag by the DMAACK signal occurs simultaneously with underflow, status flag clearing takes priority.

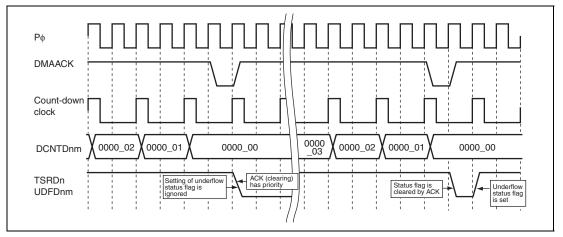


Figure 13.78 Conflict between Setting of Underflow Status Flag and Clearing by DMAACK

#### TODnmB Output at Occurrence of Down Counter Start Trigger When Down Counter **(6)** Value is H'0000 00

The TODnmB output (one-shot pulse) is not started by down counter underflow.

#### **(7)** TODnmB Output at Simultaneous Occurrence of Start Trigger and Stop Trigger for **Down Counter**

When the down counter start trigger occurs simultaneously with the down counter stop trigger, the down counter is cleared to 0 by the stop trigger. In this case, the one-shot pulse output on the TODnmB pin is not started.

Similar to the above case, when the down counter start trigger occurs and then the down counter stop trigger occurs before the first count-down clock is input, the down counter is cleared to 0 without being decremented even once and the one-shot pulse output on the TODnmB pin is not started.

#### **(8)** Conflict between Down Counter Start Trigger and Underflow

When the down counter start trigger occurs simultaneously with underflow, DCNTDnm remains halted at the value of H'0000 00 (waveforms in the middle of figure 13.79). If the down counter had been in process of decrementing, the TODnmB output is turned off by underflow (waveforms in the left half of figure 13.79). If compare match A occurs while the down counter is halted (DCNTDnm = H'0000 00), the TODnmB output is kept negated (waveforms in the right half of figure 13.79). In any case, the underflow status flag is set at the same time the down counter start trigger is detected or in synchronization with the first count-down clock after trigger detection.

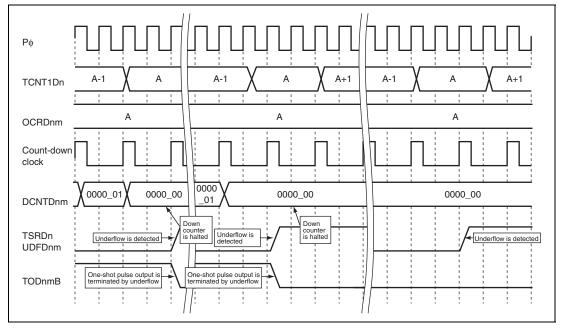


Figure 13.79 Conflict between Compare Match A and Underflow

#### 13.36.7 Conflict between Timer B and Timer D

Conflicts between the counter clearing request from timer B and TCNT1Dn/TCNT2Dn are described here.

#### (1) Conflict between TCNT1Dn/TCNT2Dn Counter Clearing and Compare Match

Operation when clearing of the TCNT1Dn/TCNT2Dn counter by a counter clearing signal from timer B occurs simultaneously with compare match is shown below. The waveforms in the left half of figure 13.80 show a case in which the counter is cleared prior to compare match. On the other hand, the waveforms in the right half of figure 13.80 show a case in which the counter is cleared simultaneously with compare match.

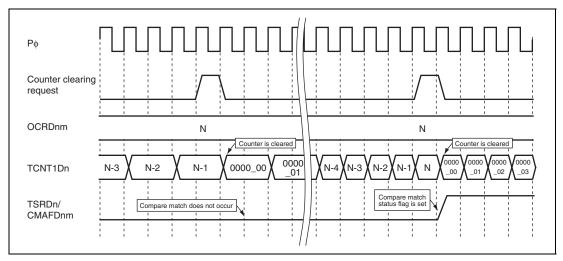


Figure 13.80 Conflict between Counter Clearing by Timer B and Compare Match

Table 13.40 Resources Related to Conflict Operation between Counter Clearing and Compare Match

| Timer   | Counter | Counter Clearing Source  | Compare Match<br>Register | Status  |
|---------|---------|--------------------------|---------------------------|---------|
| Timer D | TCNT1Dn | TCNT1Dn/TCNT2Dn clearing | OCRDnm                    | CMFADnm |
|         | TCNT2Dn | request from timer B     | GRDnm                     | CMFBDnm |

# (2) Conflict between Writing to TCNT1Dn/TCNT2Dn Counter and Counter Clearing by Timer B

When a write to TCNT1Dn/TCNT2Dn occurs simultaneously with a counter clearing signal from timer B, the counter is not cleared but writing to the counter is performed (waveforms in the left half of figure 13.81). The waveforms in the right half of figure 13.81 show a case in which writing to TCNT1Dn is one  $P\phi$  cycle later. This is the same for TCNT2Dn.

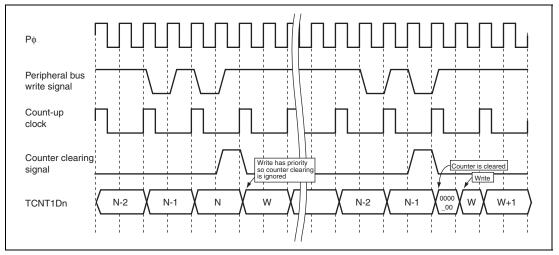


Figure 13.81 Conflict between Writing to TCNT1Dn and Counter Clearing

#### Conflict between TCNT1Dn/TCNT2Dn Counter Overflow and Counter Clearing by **(3)** Timer B

When TCNT1Dn overflow occurs simultaneously with clearing of TCNT1Dn from timer B, the counter value is cleared to H'0000 00 by the counter clearing signal. In this case, the overflow status flag is not set (only for when C1CEDn = 1). The same applies to TCNT2Dn overflow.

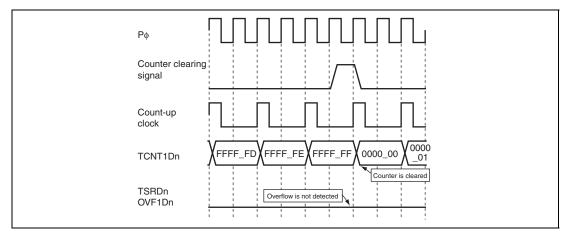


Figure 13.82 Conflict between Counter Clearing and Overflow

#### **(4)** Conflict between TCNT1Dn Clearing by Clearing Signal from Timer B and Input Capture to OSBRDn

Operation when clearing of the TCNT1Dn counter by timer B occurs simultaneously with input capture to an offset base register is shown below. When capture and counter clearing occur simultaneously, the counter value before clearing is captured in OSBRDn. At the same time, the TCNT1Dn counter is cleared (waveforms in the left half of figure 13.83).

The waveforms in the right half of figure 13.83 show a case in which capture is performed one P $\phi$ cycle after the counter has been cleared and the counter value after clearing is captured in OSBRDn.

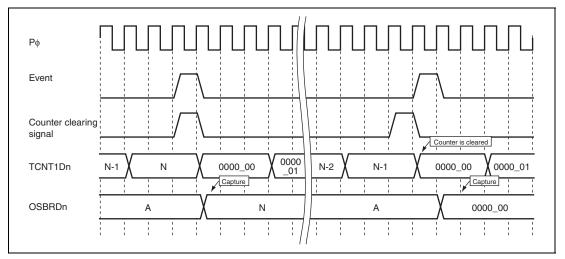


Figure 13.83 Conflict between TCNT1Dn Counter Clearing and Input Capture to OSBRDn

#### **Compare-Match Operation Specification**

The compare-match operation specification depends on each timer. Theoperation specifications can be grouped into three types, according to timing of compare-match occurrence and detection condition.

#### Type 1

A compare match occurs on the next P $\phi$  cycle after the value of a timer counter has reached the value set in the corresponding compare match register.

Detection of compare-match is performed on each Pφ cycle.

Detection of compare-match is also performed on writing to the timer counter or comparematch register.

#### Type 2

A compare match occurs on the next  $P\phi$  cycle after the value of a timer counter has reached the value set in the corresponding compare match register.

Detection of compare-match is performed on each  $P\phi$  cycle on which timer counters are incremented or decremented.

Detection of compare-match is not performed on writing to the timer counter or comparematch register.

#### Type 3

A compare match occurs on the next counter clock cycle after the value of a timer counter has reached the value set in the corresponding compare match register.

Detection of compare-match is performed on each counter clock cycle.

Detection of compare-match is not performed on writing to the timer counter or comparematch register.

**Table 13.41 Compare-match Operation Specification** 

| Timer   | Counter | Compare Match Register | Compare-match type |
|---------|---------|------------------------|--------------------|
| Timer B | TCNTB0  | OCRB0                  | Type 3             |
|         | TCNTB1  | OCRB1                  | Type 1             |
|         | TCNTB6  | OCRB6/OCRB7            | Type 3             |
| Timer C | TCNTCn  | GRCnm                  | Type 1             |
| Timer D | TCNT1Dn | OCRDnm                 | Type 2             |
|         | TCNT2Dn | GRDnm                  | Type 2             |
| Timer E | TCNTEnm | CYLREnm                | Type 3             |
| Timer F | ECNTAFn | GRAFn                  | Type 1             |
|         | ECNTBFn | GRBFn                  | Type 1             |
|         | ECNTCFn | GRBFn                  | Type 1             |
| Timer G | TCNTGn  | OCRGn                  | Type 1             |
| Timer H | TCNT1H  | OCR1H                  | Type 1             |
| Timer J | TCNTJn  | OCRJn                  | Type 1             |

## Section 14 Watchdog Timer (WDT)

This LSI includes the watchdog timer (WDT) which externally outputs an overflow signal (WDTOVF) on overflow of the counter when the value of the counter has not been updated because of a system malfunction. The WDT can simultaneously generate an internal reset signal for the entire LSI.

When this watchdog function is not needed, the WDT can be used as an interval timer. In the interval timer operation, an interval timer interrupt is generated at each counter overflow.

#### 14.1 Features

- Can switch between watchdog timer mode and interval timer mode.
- Outputs WDTOVF signal in watchdog timer mode
   When the counter overflows in watchdog timer mode, the WDTOVF signal is output externally. It is possible to select whether to reset the LSI internally when this happens.
- Interrupt generation in interval timer mode

  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks
   Eight clocks (Pφ × 1 to Pφ × 1/16384) that are obtained by dividing the peripheral clock can be selected.

Figure 14.1 is a block diagram of the WDT.

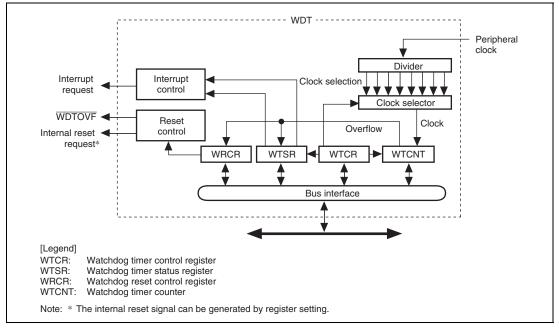


Figure 14.1 Block Diagram of WDT

### 14.2 Input/Output Pin

Table 14.1 shows the pin configuration of the WDT.

**Table 14.1 Pin Configuration** 

| Pin Name                | Symbol | I/O    | Function   |
|-------------------------|--------|--------|--|
| Watchdog timer overflow | WDTOVF | Output | Outputs the counter overflow signal in watchdog timer mode |

### 14.3 Register Descriptions

The WDT has the following registers.

**Table 14.2 Register Configuration** 

| Register Name                   | Symbol | R/W | Initial<br>Value | Address     | Access<br>Size |
|---------------------------------|--------|-----|------------------|-------------|----------------|
| Watchdog timer control register | WTCR   | R/W | H'0000           | H'FFFE 0000 | 8, 16          |
| Watchdog timer counter          | WTCNT  | R/W | H'0000           | H'FFFE 0002 | 8, 16          |
| Watchdog timer status register  | WTSR   | R/W | H'0000           | H'FFFE 0004 | 8, 16          |
| Watchdog reset control register | WRCR   | R/W | H'0000           | H'FFFE 0006 | 8, 16          |

### 14.3.1 Watchdog Timer Control Register (WTCR)

WTCR is a 16-bit readable/writable register that stores an 8-bit write key, selects the operating mode and the clock used for WTCNT counting, and enables the timer.

WTCR is initialized by a power-on reset caused by the  $\overline{RES}$  pin, a transition to the hardware standby mode, or by a WTCNT overflow in watchdog timer mode.

WTCR must be written in words. When changing the WT/IT, TME, and CKS bits, write H'A5 to the TCRKEY bits simultaneously. Writing a value other than H'A5 to the TCRKEY bits or writing in bytes are ignored.

WTCR can be read in words or bytes. Note that data written to the TCRKEY bits is not retained; therefore, the TCRKEY bits are always read as H'00.

Note: The method for writing to this register differs from that for other registers to prevent erroneous writes. See section 14.3.5, Notes on Register Access, for details.

| Bit           | : 15   | 14   | 13   | 12   | 11      | 10   | 9    | 8    | 7 | 6     | 5   | 4 | 3 | 2   | 1        | 0   |
|---------------|--------|------|------|------|---------|------|------|------|---|-------|-----|---|---|-----|----------|-----|
|               |        |      |      | TCRK | EY[7:0] |      |      |      | - | WT/IT | TME | - | - | (   | CKS[2:0] | ]   |
| Initial value | : 0    | 0    | 0    | 0    | 0       | 0    | 0    | 0    | 0 | 0     | 0   | 0 | 0 | 0   | 0        | 0   |
| R/W           | : R/W* | R/W* | R/W* | R/W* | R/W*    | R/W* | R/W* | R/W* | R | R/W   | R/W | R | R | R/W | R/W      | R/W |

| Bit     | Bit Name | Initial<br>Value | R/W  | Description  |
|---------|----------|------------------|------|--|
| 15 to 8 | TCRKEY   | H'00             | R/W* | WTCR Write Key Code  |
|         | [7:0]    |                  |      | These bits enable or disable changing the WT/IT, TME, and CKS bits. Since data written to these bits is not retained, these bits are always read as H'00.                                |
|         |          |                  |      | H'A5: Enables changing the WT/IT, TME, and CKS bits  |
|         |          |                  |      | Other than H'A5: Disables changing the WT/IT, TME, and CKS bits  |
| 7       | _        | 0                | R    | Reserved   |
|         |          |                  |      | This bit is always read as 0. The write value should always be 0.  |
| 6       | WT/IT    | 0                | R/W  | Timer Mode Select  |
|         |          |                  |      | Selects whether to use the WDT as a watchdog timer or an interval timer.   |
|         |          |                  |      | 0: WDT used as interval timer  |
|         |          |                  |      | 1: WDT used as watchdog timer  |
|         |          |                  |      | Note: When WTCNT overflows in watchdog timer mode, the WDTOVF signal is output externally. If this bit is modified when the WDT is running, the up-count may not be performed correctly. |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |  |   |
|--------|----------|------------------|-----|--|--|---|
| 5      | TME      | 0                | R/W | WTCNT Enable   |  |   |
|        |          |                  |     | Starts or stops WTCN   | NT counting.   |   |
|        |          |                  |     | 0: Counting disabled   |  |   |
|        |          |                  |     | Counting up stops  | and WTCNT val  | ue is stored                                    |
|        |          |                  |     | 1: Counting enabled  |  |   |
| 4, 3   | _        | All 0            | R   | Reserved   |  |   |
|        |          |                  |     | These bits are always should always be 0.  | s read as 0. The   | write value                                     |
| 2 to 0 | CKS[2:0] | 000              | R/W | Clock Select   |  |   |
|        |          |                  |     | These bits select the counting from the eighthe peripheral clock (I below is the value wh 20 MHz. The table be on condition that P¢ is | ht types obtainated by the second of the peripherate of the peripherated shows the o | ole by dividing<br>w cycle shown<br>al clock (P |
|        |          |                  |     | <b>Clock Division</b>  | Overflo  | ow cycle  |
|        |          |                  |     | Ratio  | Pφ = 20MHz   | Pφ = 40MHz                                      |
|        |          |                  |     | 000: 1 × Pφ  | 12.8 μs  | 6.4 μs  |
|        |          |                  |     | 001: 1/64 × Pφ   | 819.2 μs   | 409.6 μs  |
|        |          |                  |     | 010: 1/128 × Pφ  | 1.6 ms   | 0.8 ms  |
|        |          |                  |     | 011: 1/256 × Pφ  | 3.3 ms   | 1.65 ms   |
|        |          |                  |     | 100: 1/512 × Pφ  | 6.6 ms   | 3.3 ms  |
|        |          |                  |     | 101: 1/1024 × Pφ   | 13.1 ms  | 6.55 ms   |
|        |          |                  |     | 110: 1/4096 × Pφ   | 52.4 ms  | 26.2 ms   |
|        |          |                  |     | 111: 1/16384 × Pφ  | 209.7 ms   | 104.35 ms                                       |
|        |          |                  |     | performed corr   | ing, counting up   |   |

#### **14.3.2** Watchdog Timer Counter (WTCNT)

WTCNT is a 16-bit readable/writable register that stores an 8-bit write key and an 8-bit counter value. Setting the TME bit in WTCR to 1 starts counting on the internal clock selected by the CKS[2:0] bits in WTCR.

When a WTCNT overflow occurs, it generates a watchdog timer overflow signal (WDTOVF) in watchdog timer mode and an interval timer interrupt (ITI) in interval timer mode. WTCNT is initialized to H'0000 by a power-on reset caused by the RES pin, a transition to the hardware standby mode, or a WTCNT overflow in watchdog timer mode.

WTCNT must be written in words. When changing the TCNT bits, write H'5A to the TCNTKEY bits simultaneously. Writing a value other than H'5A to the TCNTKEY bits or writing in bytes are ignored.

WTCNT can be read in words or bytes. Note that data written to the TCNTKEY bits is not retained, therefore, the TCNTKEY bits are always read as H'00.

Note: The method for writing to this register differs from that for other registers to prevent erroneous writes. See section 14.3.5, Notes on Register Access, for details.

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

Note: \* Data written to these bits is not retained.

|         |           | Initial |      |  |
|---------|-----------|---------|------|--|
| Bit     | Bit Name  | Value   | R/W  | Description  |
| 15 to 8 | TCNTKEY   | H'00    | R/W* | WTCNT Write Key Code   |
|         | [7:0]     |         |      | These bits enable or disable changing the TCNT bits. Since data written to these bits is not retained, these bits are always read as H'00. |
|         |           |         |      | H'5A: Enables changing the TCNT bits   |
|         |           |         |      | Other than H'A5: Disables changing the TCNT bits   |
| 7 to 0  | TCNT[7:0] | H'00    | R/W  | 8-Bit Timer Counter Value  |
|         |           |         |      | An overflow is generated when these bits change from H'FF to H'00.   |

#### 14.3.3 Watchdog Timer Status Register (WTSR)

WTSR is a 16-bit readable/writable register that stores an 8-bit write key, an overflow flag in watchdog timer mode, and an overflow flag in interval timer mode.

WTSR is initialized to H'0000 by a power-on reset caused by the RES pin or a transition to the hardware standby mode. It is not initialized by a WTCNT overflow in watchdog timer mode or by an internal reset caused by a WTCNT overflow.

WTSR must be written in words. When changing the WOVF and IOVF bits, write H'A5 to the TSRKEY bits simultaneously. Writing a value other than H'A5 to the TSRKEY bits or writing in bytes are ignored.

WSTR can be read in words or bytes. Note that data written to the TSRKEY bits is not retained, therefore, the TSRKEY bits are always read as H'00.

Note: The method for writing to this register differs from that for other registers to prevent erroneous writes. See section 14.3.5, Notes on Register Access, for details.

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

Notes: 1. Data written to these bits is not retained.

2. Only 0 can be written to clear the flag. Writing 1 to the flag is invalid.

|         |          | Initial |       |  |  |   |
|---------|----------|---------|-------|--|--|---|
| Bit     | Bit Name | Value   | R/W   | Description  |  |   |
| 15 to 8 | TSRKEY   | H'00    | R/W*1 | WTCR Write Key Code  |  |   |
|         | [7:0]    |         |       | I  |  | These bits enable or disable changing the WOVF and IOVF bits. Since data written to these bits is not retained, these bits are always read as H'00. |
|         |          |         |       | H'A5: Enables changing the WOVF and IOVF bits.             |  |   |
|         |          |         |       | Other than H'A5: Disables changing the WOVF and IOVF bits. |  |   |

| Bit    | Bit Name | Initial<br>Value | R/W               | Description  |
|--------|----------|------------------|-------------------|--|
| 7      | WOVF     | 0                | R/W* <sup>2</sup> | Watchdog Timer Overflow  |
|        |          |                  |                   | Indicates that WTCNT has overflowed in watchdog timer mode. This bit is not set to 1 in interval timer mode. |
|        |          |                  |                   | 0: WTCNT has not overflowed  |
|        |          |                  |                   | 1: WTCNT has overflowed in watchdog timer mode   |
|        |          |                  |                   | [Clearing condition]   |
|        |          |                  |                   | When 0 is written to this bit after reading it as 1  |
| 6 to 4 | _        | All 0            | R                 | Reserved   |
|        |          |                  |                   | These bits are always read as 0. The write value should always be 0.   |
| 3      | IOVF     | 0                | R/W* <sup>2</sup> | Interval Timer Overflow  |
|        |          |                  |                   | Indicates that WTCNT has overflowed in interval timer mode. This bit is not set to 1 in watchdog timer mode. |
|        |          |                  |                   | 0: WTCNT has not overflowed  |
|        |          |                  |                   | 1: WTCNT has overflowed in interval timer mode   |
|        |          |                  |                   | [Clearing condition]   |
|        |          |                  |                   | When 0 is written to this bit after reading it as 1  |
| 2 to 0 | _        | All 0            | R                 | Reserved   |
|        |          |                  |                   | These bits are always read as 0. The write value should always be 0.   |

Notes: 1. Data written to these bits is not retained.

2. Only 0 can be written to clear the flag. Writing 1 to the flag is invalid.

#### 14.3.4 Watchdog Reset Control Register (WRCR)

WRCR is a 16-bit readable/writable register that stores an 8-bit write key and enables or disables an internal reset caused by a WTCNT overflow.

WRCR is initialized to H'0000 by a power-on reset caused by the  $\overline{RES}$  pin or a transition to the hardware standby mode. It is not initialized by a WTCNT overflow in watchdog timer mode or by an internal reset caused by a WTCNT overflow.

WRCR must be written in words. When changing the RSTE bit, write H'5A to the RCRKEY bits simultaneously. Writing a value other than H'5A to the RCRKEY bits or writing in bytes are ignored.

WRCR can be read in words or bytes. Note that data written to the RCRKEY bits is not retained, therefore, the RCRKEY bits are always read as H'00.

Note: The method for writing to WRCR differs from that for other registers to prevent erroneous writes. See section 14.3.5, Notes on Register Access, for details.

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

Note: \* Data written to these bits is not retained.

|                                |                     | Initial |  |   |  |  |  |
|--------------------------------|---------------------|---------|--|---|--|--|--|
| Bit                            | Bit Name            | Value   | R/W  | /W* WRCR Write Key Code  These bits enable or disable changing the RSTE by  |  |  |  |
| 15 to 8 RCRKEY H'00 R/W* [7:0] | WRCR Write Key Code |         |  |   |  |  |  |
|                                | [7:0]               |         | These bits enable or disa<br>Since data written to thes<br>bits are always read as H | These bits enable or disable changing the RSTE bit. Since data written to these bits is not retained, these bits are always read as H'00. |  |  |  |
|                                |                     |         |  | H'5A: Enables changing the RSTE bit.  |  |  |  |
|                                |                     |         |  | Other than H'A5: Disables changing the RSTE bit.  |  |  |  |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7      | RSTE     | 0                | R/W | Reset Enable  |
|        |          |                  |     | Selects whether to generate a signal to initialize the LSI internally if WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. |
|        |          |                  |     | 0: This LSI is not initialized when WTCNT overflows*  |
|        |          |                  |     | 1: This LSI is initialized when WTCNT overflows   |
|        |          |                  |     | Note: This LSI is not initialized internally, but WTCNT and WTCR are initialized by an overflow of WTCNT.   |
| 6 to 0 | _        | All 0            | R   | Reserved  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.  |

#### **Notes on Register Access** 14.3.5

The watchdog timer counter (WTCNT), watchdog timer control register (WTCR), watchdog timer status register (WTSR), and watchdog reset control register (WRCR) are more difficult to write to than other registers. The procedures for reading or writing to these registers are given below.

#### Writing to WTCNT, WTCR, WTSR, and WRCR **(1)**

These registers must be written by a word transfer instruction. They cannot be written by a byte transfer instruction.

When writing to WTCNT and WRCR, set the upper byte to H'5A and the lower byte to the write data as shown in figure 14.2 and then transfer data. When writing to WTCR and WTSR, set the upper byte to H'A5 and the lower byte to the write data and then transfer data. This transfer procedure writes the lower byte data to WTCNT, WTCR, WTSR, or WRCR.

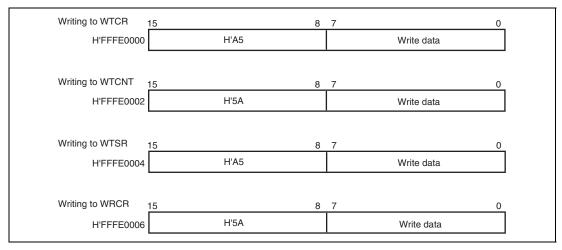


Figure 14.2 Writing to WTCNT, WTCR, WTSR, and WRCR

#### Reading from WTCNT, WTCR, WTSR, and WRCR **(2)**

These registers are read in a method similar to other registers. Both byte and word transfer instructions can be used.

### 14.4 WDT Usage

#### 14.4.1 Using WDT in Watchdog Timer Mode

- 1. Set the WT/TT bit in WTCR to 1 (watchdog timer mode). Set the CKS[2:0] bits in WTCR (the type of counting clock), the RSTE bit in WRCR (whether to initialize this LSI on a WTCNT overflow), and WTCNT to the initial value of the counter.
- 2. Set the TME bit in WTCR to 1 to start counting in watchdog timer mode.
- 3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
- 4. When the counter overflows, the WDT sets the WOVF bit in WTCR to 1, and the WDTOVF signal is output externally (figure 14.3). The WDTOVF signal can be used to initialize the system. The WDTOVF signal is output for 64 × Pφ clock cycles.
- If the RSTE bit in WRCR is set to 1, a signal to initialize the inside of this LSI can be generated simultaneously with the WDTOVF signal. The internal reset signal is output for 128 × Pφ clock cycles.
- 6. When a WTCNT overflow reset is generated simultaneously with a reset input on the RES pin, the RES pin reset takes priority, and the WOVF bit in WTSR is cleared to 0.

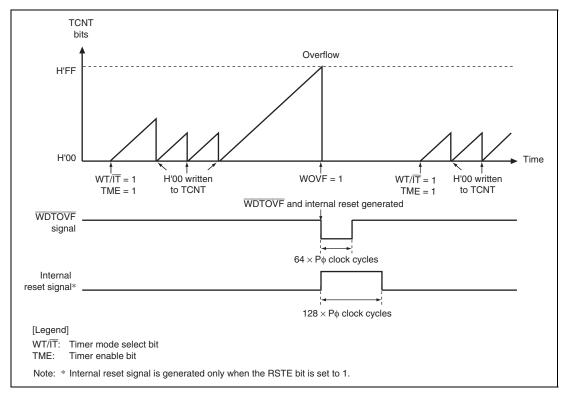


Figure 14.3 Operation in Watchdog Timer Mode

### 14.4.2 Using WDT in Interval Timer Mode

When using the WDT in interval timer mode, an interval timer interrupt (ITI) is generated every overflow of the counter. This enables interrupts to be generated at specified periods.

- 1. Clear the WT/IT bit in WTCR to 0. Set the CKS[2:0] bits in WTCR (the type of counting clock) and the TCNT bits in WTCNT to the initial value of the counter.
- 2. Set the TME bit in WTCR to 1 to start the counting in interval timer mode.
- 3. When the counter overflows, the WDT sets the IOVF bit in WTCR to 1 and an interval timer interrupt (ITI) is requested to the INTC. The counter then resumes counting.

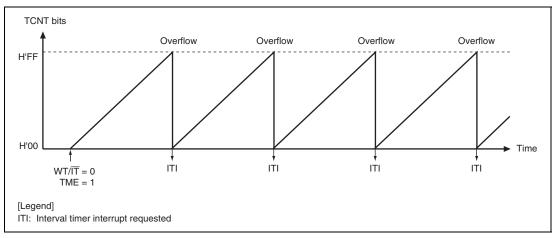


Figure 14.4 Operation in Interval Timer Mode

### 14.5 Usage Notes

Pay attention to the following points when using the WDT in either the interval or watchdog timer mode.

#### 14.5.1 Timer Error

It takes one cycle of the counting clock until WTCNT counts up for the first time after the TME bit in WTCR is set to start, meaning that it takes one cycle of the  $P\phi$  clock at a minimum and one cycle of the divided  $P\phi$  clock at worst (the division ratio is selected by the CKS[2:0] bits). The timer error is the duration from when leaving the power-on reset state to the first counting-up timing after the TME bit is set. The timing of subsequent counting up depends on the selected frequency division ratio.

This also applies to the timing of the first counting up after WTCNT has been written to during timer operation.

#### 14.5.2 Changing of Division Ratio

If the CKS[2:0] bits in WTCR are changed during WTCNT counting, correct operation may not be guaranteed. Change these bits after stopping the WDT (after clearing the TME bit to 0).

### 14.5.3 Switching between Watchdog and Interval Timer Modes

If the watchdog and interval timer modes are switched during WDT operation, correct operation may not be guaranteed. Switch timer modes after stopping the WDT (after clearing the TME bit to 0).

### 14.5.4 System Reset by WDTOVF Signal

If the WDTOVF signal is input on the RES pin of this LSI, this LSI cannot be initialized correctly.

Avoid input of the  $\overline{WDTOVF}$  signal to the  $\overline{RES}$  pin of this LSI through glue logic circuits. To initialize the entire system with the  $\overline{WDTOVF}$  signal, use the circuit shown in figure 14.5.

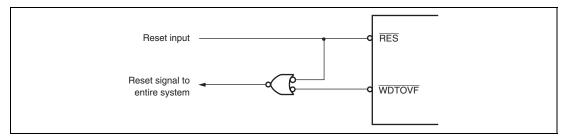


Figure 14.5 Example of System Reset Circuit Using WDTOVF Signal

## Section 15 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer (CMT) consisting of a two-channel 16-bit timer. The CMT has a16-bit counter, and can generate interrupts at set intervals.

#### 15.1 **Features**

- Independent selection of four counter input clocks at two channels Any of four internal clocks (P $\phi$ /8, P $\phi$ /32, P $\phi$ /128, and P $\phi$ /512) can be selected.
- Selection of DMA transfer request or interrupt request generation on compare match by DMAC setting

Figure 15.1 shows a block diagram of CMT.

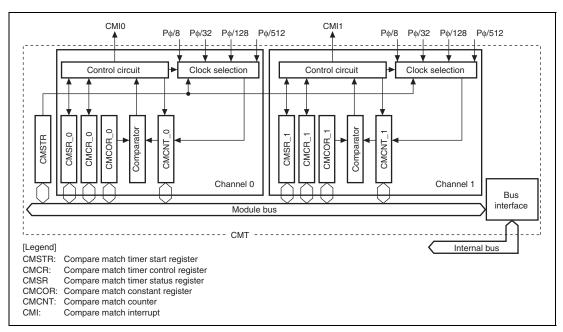


Figure 15.1 Block Diagram of CMT

## 15.2 Register Descriptions

The CMT has the following registers.

**Table 15.1 Register Configuration** 

| Channel | Register Name                          | Abbreviation | R/W    | Initial<br>Value | Address    | Access<br>Size |
|---------|--|--------------|--------|------------------|------------|----------------|
| Common  | Compare match timer start register     | CMSTR        | R/W    | H'0000           | H'FFFEC000 | 16             |
| 0       | Compare match timer control register_0 | CMCR_0       | R/W    | H'00             | H'FFFEC010 | 8              |
|         | Compare match timer status register_0  | CMSR_0       | R/(W)* | H'00             | H'FFFEC011 | 8              |
|         | Compare match counter_0                | CMCNT_0      | R/W    | H'0000           | H'FFFEC012 | 16             |
|         | Compare match constant register_0      | CMCOR_0      | R/W    | H'FFFF           | H'FFFEC014 | 16             |
| 1       | Compare match timer control register_1 | CMCR_1       | R/W    | H'00             | H'FFFEC020 | 8              |
|         | Compare match timer status register_1  | CMSR_1       | R/(W)* | H'00             | H'FFFEC021 | 8              |
|         | Compare match counter_1                | CMCNT_1      | R/W    | H'0000           | H'FFFEC022 | 16             |
|         | Compare match constant register_1      | CMCOR_1      | R/W    | H'FFFF           | H'FFFEC024 | 16             |

### 15.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether or not the compare match counter (CMCNT) operates.

CMSTR is initialized to H'0000 by a power-on reset or in standby mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1    | 0    |
|----------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|------|------|
|                | -  | -  | -  | -  | -  | -  | - | - | - | - | - | - | - | - | STR1 | STR0 |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0    |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R | R | R | R/W  | R/W  |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 15 to 2 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 1       | STR1     | 0                | R/W | Count Start 1  |
|         |          |                  |     | Specifies whether or not compare match counter 1 operates.           |
|         |          |                  |     | 0: Stops counting by CMCNT_1*  |
|         |          |                  |     | 1: Starts counting by CMCNT_1  |
| 0       | STR0     | 0                | R/W | Count Start 0  |
|         |          |                  |     | Specifies whether or not compare match counter 0 operates.           |
|         |          |                  |     | 0: Stops counting by CMCNT_0*  |
|         |          |                  |     | 1: Starts counting by CMCNT_0  |

Note: \* The value in CMCNT is retained when counting stops.

### 15.2.2 Compare Match Timer Control Register (CMCR)

CMCR is an 8-bit register that enables interrupts and DMA transfer requests, and selects the counter input clock.

CMCR is initialized to H'00 by a power-on reset or entry into standby mode.

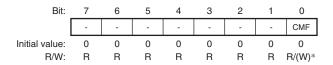
| Bit:           | 7 | 6    | 5 | 4 | 3 | 2 | 1   | 0      |
|----------------|---|------|---|---|---|---|-----|--------|
|                | - | CMIE | - | - | - | - | CKS | S[1:0] |
| Initial value: | 0 | 0    | 0 | 0 | 0 | 0 | 0   | 0      |
| R/W:           | R | R/W  | R | R | R | R | R/W | R/W    |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7      | _        | 0                | R   | Reserved   |
|        |          |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 6      | CMIE     | 0                | R/W | Compare Match Interrupt Enable   |
|        |          |                  |     | Enables or disables compare match interrupt (CMI) generation when CMCNT and CMCOR values match (CMF = 1).  |
|        |          |                  |     | 0: Disables the compare match interrupt (CMI)  |
|        |          |                  |     | 1: Enables the compare match interrupt (CMI)   |
| 5 to 2 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 1, 0   | CKS[1:0] | 00               | R/W | Clock Select   |
|        |          |                  |     | These bits select the clock to be input to CMCNT among four types of internal clocks, which are obtained by dividing the peripheral clock (P $\phi$ ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS1 and CKS0. |
|        |          |                  |     | 00: Pφ/8   |
|        |          |                  |     | 01: Pφ/32  |
|        |          |                  |     | 10: P  |
|        |          |                  |     | 11: P  þ/512   |

#### 15.2.3 Compare Match Timer Status Register (CMSR)

CMSR is an 8-bit register that indicates compare match generation.

CMSR is initialized to H'00 by a power-on reset or in standby mode.



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

| Bit    | Bit Name | Initial<br>Value | R/W     | Description  |
|--------|----------|------------------|---------|--|
| 7 to 1 | _        | All 0            | R       | Reserved   |
|        |          |                  |         | These bits are always read as 0. The write value should always be $0. \  \  $  |
| 0      | CMF      | 0                | R/(W)*1 | Compare Match Flag   |
|        |          |                  |         | Indicates whether or not the values of CMCNT and CMCOR match.  |
|        |          |                  |         | 0: CMCNT and CMCOR values do not match   |
|        |          |                  |         | [Clearing condition]   |
|        |          |                  |         | <ul> <li>When 0 is written to CMF after reading CMF = 1</li> <li>1: CMCNT and CMCOR values match*<sup>2</sup></li> </ul> |

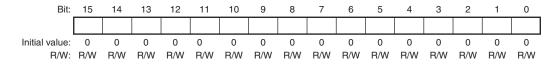
Notes: 1. Only 0 can be written to clear the flag after 1 is read.

2. The CMF bit is also set to 1 when the values in CMCNT and CMCOR match by writing the same value to CMCNT as in CMCOR.

### 15.2.4 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS1 and CKS0 in CMCR, and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock. When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMSR is set to 1.

CMCNT is initialized to H'0000 by a power-on reset or in standby mode.



#### 15.2.5 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power-on reset or in standby mode.

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

### 15.3 Operation

#### 15.3.1 Interval Count Operation

When an internal clock is selected with the CKS1 and CKS0 bits in CMCR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'0000 and the CMF flag in CMSR is set to 1. When the CMIE bit in CMCR is set to 1 at this time, a compare match interrupt (CMI) is requested. CMCNT then starts counting up again from H'0000.

Figure 15.2 shows the operation of the compare match counter.

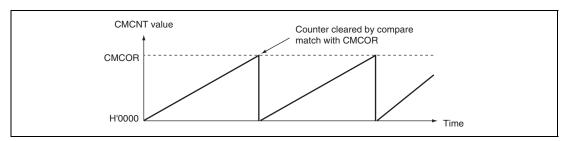


Figure 15.2 Counter Operation

#### 15.3.2 CMCNT Count Timing

One of four clocks ( $P\phi/8$ ,  $P\phi/32$ ,  $P\phi/128$ , and  $P\phi/512$ ) obtained by dividing the peripheral clock ( $P\phi$ ) can be selected with the CKS[1:0] bits in CMCR. Figure 15.3 shows the timing.

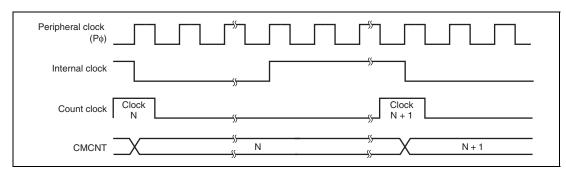


Figure 15.3 Count Timing

### 15.4 Interrupts

#### 15.4.1 Interrupt Sources and DMA Transfer Requests

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt. When both the interrupt request flag (CMF) and the interrupt enable bit (CMIE) are set to 1, the corresponding interrupt request is output. When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 8, Interrupt Controller (INTC).

Clear the CMF bit to 0 by the user exception handling routine. If this operation is not carried out, another interrupt will be generated. The direct memory access controller (DMAC) can be set to be activated when a compare match interrupt is requested. In this case, an interrupt is not issued to the CPU. If the setting to activate the DMAC has not been made, an interrupt request is sent to the CPU. The CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

#### 15.4.2 Timing of Compare Match Flag Setting

When CMCOR and CMCNT match, a compare match signal is generated and the CMF bit in CMSR is set to 1. The compare match signal is generated in the last state in which the values match (when the CMCNT value is updated to H'0000). That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 15.4 shows the timing of CMF bit setting.

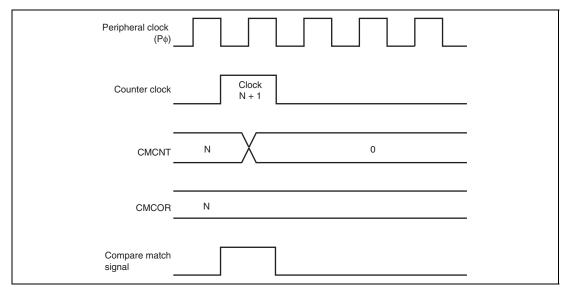


Figure 15.4 Timing of CMF Setting

#### 15.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMSR is cleared by first, reading as 1 then writing to 0. However, in the case of the DMAC being activated, the CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

Note that the CMF bit is cleared regardless of the state of counter operation. Even when counting is not in progress, the CMF bit is cleared by writing to the CMSR register by the CPU or upon generation of an ACK signal from the DMAC.

### 15.5 Usage Notes

#### 15.5.1 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated in the T2 cycle while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 15.5 shows the timing to clear the CMCNT counter.

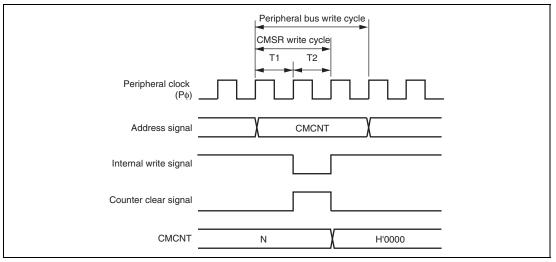


Figure 15.5 Conflict between Write and Compare Match Processes of CMCNT

#### 15.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in words, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 15.6 shows the timing to write to CMCNT in words.

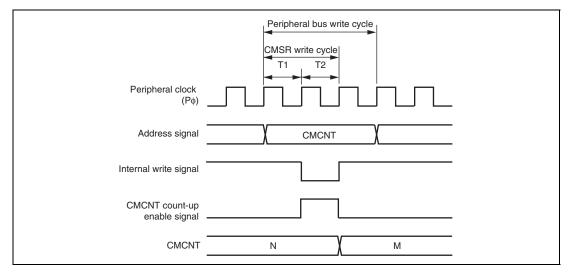


Figure 15.6 Conflict between Word-Write and Count-Up Processes of CMCNT

## 15.5.3 Conflict between Setting of Compare-Match Flag and Clearing by the CPU

If, while the compare-match flag is set, clearing of the flag by the CPU (during T2 cycle within CMSR register write cycle), that is, writing a 0 to the flag after reading a 1 from it, and setting of the flag on a compare match coincide, setting of the flag is given priority. If a compare-match occurs during the period between reading a 1 from and writing a 0 to the flag, clearing by writing a 0 will not be done.

Figure 15.7 shows the timing explained above. The left shows the case where setting and clearing the flag coincide, and the right shows the case where a compare-match occurs after reading a 1 from the flag and before writing a 0 to it.

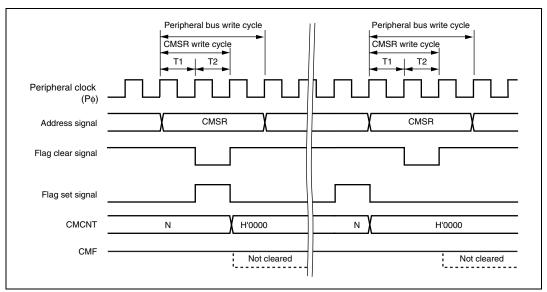


Figure 15.7 Conflict between Compare-Match Flag Setting and Clearing by the CPU

# Section 16 Serial Communications Interface (SCI)

The SCI can handle both asynchronous and clock synchronous serial communications.

## 16.1 Features

- Choice of asynchronous or clock synchronous serial communications mode
- Asynchronous mode:
  - Serial data communications are performed by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communications interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are twelve selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: Break is detected by reading the RxD pin level directly when a framing error occurs.
- Clock synchronous mode:
  - Serial data communications are synchronized with a clock signal. The SCI can communicate with other chips having a clock synchronous communications function.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communications: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so highspeed 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 in clock synchronous mode: From either baud rate generator (internal clock) or SCK pin (external clock)
- Four types of interrupts: There are four interrupt sources, transmit-data-empty, transmit end, receive-data-full, and receive error interrupts, and each interrupt can be requested independently. The automotive direct memory access controller (A-DMAC) can be activated by the transmit-data-empty interrupt or receive-data-full interrupt to transfer data.

Figure 16.1 shows a block diagram of the SCI.

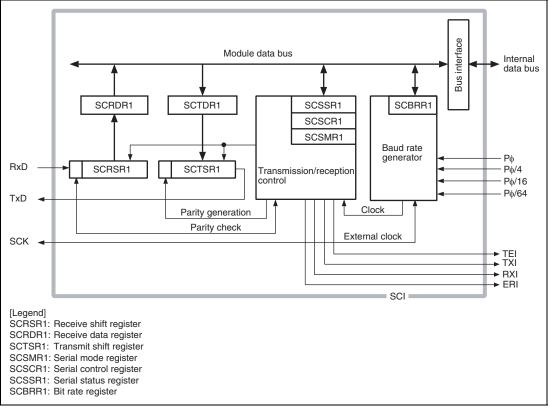


Figure 16.1 Block Diagram of SCI

#### 16.2 **Input/Output Pins**

Table 16.1 shows the pin configuration of the SCI.

**Table 16.1 Pin Configuration** 

| Channel | Pin Name*          | Symbol | I/O    | Function             |
|---------|--------------------|--------|--------|----------------------|
| A       | Serial clock pins  | SCK_A  | I/O    | Clock input/output   |
|         | Receive data pins  | RxD_A  | Input  | Receive data input   |
|         | Transmit data pins | TxD_A  | Output | Transmit data output |
| В       | Serial clock pins  | SCK_B  | I/O    | Clock input/output   |
|         | Receive data pins  | RxD_B  | Input  | Receive data input   |
|         | Transmit data pins | TxD_B  | Output | Transmit data output |
| С       | Serial clock pins  | SCK_C  | I/O    | Clock input/output   |
|         | Receive data pins  | RxD_C  | Input  | Receive data input   |
|         | Transmit data pins | TxD_C  | Output | Transmit data output |
| D       | Serial clock pins  | SCK_D  | I/O    | Clock input/output   |
|         | Receive data pins  | RxD_D  | Input  | Receive data input   |
|         | Transmit data pins | TxD_D  | Output | Transmit data output |
| E       | Serial clock pins  | SCK_E  | I/O    | Clock input/output   |
|         | Receive data pins  | RxD_E  | Input  | Receive data input   |
|         | Transmit data pins | TxD_E  | Output | Transmit data output |

Notes: 1. These pins serve as serial pins if the SCI operation modes are set appropriately by the TE, RE, and CKE1 bits in SCSCR1 and the C/A bit in SCSMR1.

<sup>2.</sup> Pin names SCK, RxD, and TxD are used in the description for all channels, omitting the channel designation.

## 16.3 Register Description

The SCI has the following registers. These registers specify the asynchronous or clock synchronous mode, data format, bit rate, and control the transmitter or the receiver. For details on register addresses and register states during each processing state, refer to section 31, List of Registers.

**Table 16.2 Register Configuration** 

| Channel | Register Name           | Abbreviation* <sup>1</sup> | R/W     | Initial<br>Value | Address    | Access<br>Size |
|---------|-------------------------|----------------------------|---------|------------------|------------|----------------|
| A       | Serial mode register    | SCSMR1A                    | R/W     | H'00             | H'FFFF8000 | 8              |
|         | Bit rate register       | SCBRR1A                    | R/W     | H'FF             | H'FFFF8004 | 8              |
|         | Serial control register | SCSCR1A                    | R/W     | H'00             | H'FFFF8008 | 8              |
|         | Transmit data register  | SCTDR1A                    | R/W     | H'FF             | H'FFFF800C | 8              |
|         | Serial status register  | SCSSR1A                    | R/(W)*2 | H'84             | H'FFFF8010 | 8              |
|         | Receive data register   | SCRDR1A                    | R       | H'00             | H'FFFF8014 | 8              |
| В       | Serial mode register    | SCSMR1B                    | R/W     | H'00             | H'FFFF8800 | 8              |
|         | Bit rate register       | SCBRR1B                    | R/W     | H'FF             | H'FFFF8804 | 8              |
|         | Serial control register | SCSCR1B                    | R/W     | H'00             | H'FFFF8808 | 8              |
|         | Transmit data register  | SCTDR1B                    | R/W     | H'FF             | H'FFFF880C | 8              |
|         | Serial status register  | SCSSR1B                    | R/(W)*2 | H'84             | H'FFFF8810 | 8              |
|         | Receive data register   | SCRDR1B                    | R       | H'00             | H'FFFF8814 | 8              |
| С       | Serial mode register    | SCSMR1C                    | R/W     | H'00             | H'FFFF9000 | 8              |
|         | Bit rate register       | SCBRR1C                    | R/W     | H'FF             | H'FFFF9004 | 8              |
|         | Serial control register | SCSCR1C                    | R/W     | H'00             | H'FFFF9008 | 8              |
|         | Transmit data register  | SCTDR1C                    | R/W     | H'FF             | H'FFFF900C | 8              |
|         | Serial status register  | SCSSR1C                    | R/(W)*2 | H'84             | H'FFFF9010 | 8              |
|         | Receive data register   | SCRDR1C                    | R       | H'00             | H'FFFF9014 | 8              |

| Channel | Register Name           | Abbreviation* <sup>1</sup> | R/W     | Initial<br>Value | Address    | Access<br>Size |
|---------|-------------------------|----------------------------|---------|------------------|------------|----------------|
| D       | Serial mode register    | SCSMR1D                    | R/W     | H'00             | H'FFFF9800 | 8              |
|         | Bit rate register       | SCBRR1D                    | R/W     | H'FF             | H'FFFF9804 | 8              |
|         | Serial control register | SCSCR1D                    | R/W     | H'00             | H'FFFF9808 | 8              |
|         | Transmit data register  | SCTDR1D                    | R/W     | H'FF             | H'FFFF980C | 8              |
|         | Serial status register  | SCSSR1D                    | R/(W)*2 | H'84             | H'FFFF9810 | 8              |
|         | Receive data register   | SCRDR1D                    | R       | H'00             | H'FFFF9814 | 8              |
| E       | Serial mode register    | SCSMR1E                    | R/W     | H'00             | H'FFFFA000 | 8              |
|         | Bit rate register       | SCBRR1E                    | R/W     | H'FF             | H'FFFFA004 | 8              |
|         | Serial control register | SCSCR1E                    | R/W     | H'00             | H'FFFFA008 | 8              |
|         | Transmit data register  | SCTDR1E                    | R/W     | H'FF             | H'FFFFA00C | 8              |
|         | Serial status register  | SCSSR1E                    | R/(W)*2 | H'84             | H'FFFFA010 | 8              |
|         | Receive data register   | SCRDR1E                    | R       | H'00             | H'FFFFA014 | 8              |

Notes: 1. Register names and abbreviations are used in the description for all channels, omitting the channel designation.

2. Writing only 0 is enabled to clear the flag.

## 16.3.1 Receive Shift Register (SCRSR1)

SCRSR1 receives serial data. Data input at the RxD pin is loaded into SCRSR1 in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to SCRDR1. The CPU cannot read or write to SCRSR1 directly.



## 16.3.2 Receive Data Register (SCRDR1)

SCRDR1 is a register that stores serial receive data. After receiving one byte of serial data, the SCI transfers the received data from the receive shift register (SCRSR1) into SCRDR1 for storage and completes operation. After that, SCRSR1 is ready to receive data.

Since SCRSR1 and SCRDR1 work as a double buffer in this way, data can be received continuously.

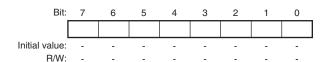
SCRDR1 is a read-only register and cannot be written to by the CPU.

SCRDR1 is initialized to H'00 by a power on reset or in standby mode.



## 16.3.3 Transmit Shift Register (SCTSR1)

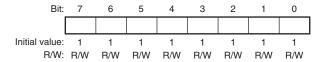
SCTSR1 transmits serial data. The SCI loads transmit data from the transmit data register (SCTDR1) into SCTSR1, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from SCTDR1 into SCTSR1 and starts transmitting again. If the TDRE flag in the serial status register (SCSSR1) is set to 1, the SCI does not transfer data from SCTDR1 to SCTSR1. The CPU cannot read or write to SCTSR1 directly.



## 16.3.4 Transmit Data Register (SCTDR1)

SCTDR1 is an 8-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (SCTSR1) is empty, it moves transmit data written in the SCTDR1 into SCTSR1 and starts serial transmission. If the next transmit data has been written to SCTDR1 during serial transmission from SCTSR1, the SCI can transmit data continuously. SCTDR1 can always be written or read to by the CPU.

SCTDR1 is initialized to H'FF by a power on reset or in standby mode.



## 16.3.5 Serial Mode Register (SCSMR1)

SCSMR1 is an 8-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR1.

SCSMR1 is initialized to H'00 by a power on reset or in standby mode.



| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 7   | C/A      | 0                | R/W | Communication Mode  |
|     |          |                  |     | Selects whether the SCI operates in asynchronous or clock synchronous mode. |
|     |          |                  |     | 0: Asynchronous mode  |
|     |          |                  |     | 1: Clock synchronous mode   |

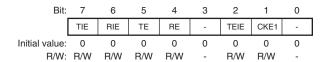
| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 6   | CHR      | 0                | R/W | Character Length   |
|     |          |                  |     | Selects 7-bit or 8-bit data in asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR bit setting. When 7-bit data is selected, the MSB (bit 7) of the transmit data register (SCTDR1) is not transmitted.  |
|     |          |                  |     | 0: 8-bit data  |
|     |          |                  |     | 1: 7-bit data  |
| 5   | PE       | 0                | R/W | Parity Enable  |
|     |          |                  |     | Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting.  |
|     |          |                  |     | 0: Parity bit not added or checked   |
|     |          |                  |     | 1: Parity bit added and checked*   |
|     |          |                  |     | Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/E) setting. Receive data parity is checked according to the even/odd (O/E) mode setting.   |
| 4   | O/E      | 0                | R/W | Parity Mode  |
|     |          |                  |     | Selects even or odd parity when parity bits are added and checked. The O/Ē setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/Ē setting is ignored in clock synchronous mode, or in asynchronous mode when parity addition and checking is disabled. |
|     |          |                  |     | 0: Even parity   |
|     |          |                  |     | 1: Odd parity  |
|     |          |                  |     | If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined.   |
|     |          |                  |     | If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 3    | STOP     | 0                | R/W | Stop Bit Length   |
|      |          |                  |     | Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clock synchronous mode because no stop bits are added.  |
|      |          |                  |     | 0: One stop bit*1   |
|      |          |                  |     | 1: Two stop bits* <sup>2</sup>  |
|      |          |                  |     | When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character.   |
|      |          |                  |     | Notes: 1. When transmitting, a single 1-bit is added at the end of each transmitted character.  |
|      |          |                  |     | <ol><li>When transmitting, two 1 bits are added at the<br/>end of each transmitted character.</li></ol>   |
| 2    | _        | 0                | _   | Reserved  |
|      |          |                  |     | This bit is always read as 0. The write value should always be 0.   |
| 1, 0 | CKS[1:0] | 00               | R/W | Clock Select 1 and 0  |
|      |          |                  |     | Select the internal clock source of the on-chip baud rate generator. Four clock sources are available. P $\phi$ , P $\phi$ /4, P $\phi$ /16 and P $\phi$ /64. For further information on the clock source, bit rate register settings, and baud rate, see section 16.3.8, Bit Rate Register (SCBRR1). |
|      |          |                  |     | 00: Pφ  |
|      |          |                  |     | 01: P <sub>\$\phi\$</sub> /4  |
|      |          |                  |     | 10: Pφ/16   |
|      |          |                  |     | 11: P   |
|      |          |                  |     | Note: Pφ: Peripheral clock  |

## 16.3.6 Serial Control Register (SCSCR1)

SCSCR1 is an 8-bit register that enables or disables SCI transmission/reception and interrupt requests and selects the transmit/receive clock source. The CPU can always read and write to SCSCR1.

SCSCR1 is initialized to H'00 by a power on reset or in standby mode.



| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 7   | TIE      | 0                | R/W | Transmit Interrupt Enable   |
|     |          |                  |     | Enables or disables a transmit-data-empty interrupt (TXI) to be issued when the TDRE flag in the serial status register (SCSSR1) is set to 1 after serial transmit data is sent from the transmit data register (SCTDR1) to the transmit shift register (SCTSR1). |
|     |          |                  |     | 0: Transmit-data-empty interrupt request (TXI) is disabled*   |
|     |          |                  |     | 1: Transmit-data-empty interrupt request (TXI) is enabled   |
|     |          |                  |     | Note: * TXI can be canceled by clearing the TDRE flag to 0 after reading TDRE = 1 or by clearing the TIE bit to 0. TXI can also be canceled when data is written to SCTDR1 through the A-DMAC.  |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 6   | RIE      | 0                | R/W | Receive Interrupt Enable   |
|     |          |                  |     | Enables or disables a receive-data-full interrupt (RXI) and a receive error interrupt (ERI) to be issued when the RDRF flag in SCSSR1 is set to 1 after the serial data received is transferred from the receive shift register (SCRSR1) to the receive data register (SCRDR1).  |
|     |          |                  |     | 0: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled*  |
|     |          |                  |     | Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled   |
|     |          |                  |     | Note: * RXI can be canceled by clearing the RDRF flag after reading RDRF = 1 or by clearing the RIE bit to 0. RXI can also be canceled when data is read from SCRDR1 through the A-DMAC. ERI can be canceled by clearing the FER, PER, and ORER flags after reading FER, PER, or ORER = 1 or by clearing the RIE bit to 0. |
| 5   | TE       | 0                | R/W | Transmit Enable  |
|     |          |                  |     | Enables or disables the SCI serial transmitter. To disable the transmitter and a transmit-data-full interrupt (TXI) simultaneously, clear the TE and TIE bits to 0 simultaneously.   |
|     |          |                  |     | 0: Transmitter disabled*1  |
|     |          |                  |     | 1: Transmitter enabled*2   |
|     |          |                  |     | Notes: 1. The TDRE flag in SCSSR1 is fixed at 1.   |
|     |          |                  |     | <ol> <li>Serial transmission starts after writing<br/>transmit data into SCTDR1 and clearing the<br/>TDRE flag in SCSSR1 to 0 while the<br/>transmitter is enabled. Select the transmit<br/>format in the serial mode register<br/>(SCSMR1) before setting TE to 1.</li> </ol>   |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 4   | RE       | 0                | R/W | Receive Enable  |
|     |          |                  |     | Enables or disables the SCI serial receiver.  |
|     |          |                  |     | 0: Receiver disabled*1  |
|     |          |                  |     | 1: Receiver enabled* <sup>2</sup>   |
|     |          |                  |     | Notes: 1. Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, and ORER). These flags retain their previous values. To disable the receiver and a receive-data-full interrupt (RXI) simultaneously, clear the RE and RIE bits to 0 simultaneously. |
|     |          |                  |     | <ol> <li>Serial reception starts when a start bit is<br/>detected in asynchronous mode, or<br/>synchronous clock input is detected in clock<br/>synchronous mode. Select the receive<br/>format in SCSMR1 before setting RE to 1.</li> </ol>                      |
| 3   | _        | 0                | _   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.   |
| 2   | TEIE     | 0                | R/W | Transmit End Interrupt Enable   |
|     |          |                  |     | Enables or disables a transmit end interrupt (TEI) to be issued when no valid transmit data is found in SCTDR1 during MSB data transmission.  |
|     |          |                  |     | TEI can be canceled by clearing the TEND flag to 0 (by clearing the TDRE flag in SCSSR1 to 0 after reading TDRE = 1) or by clearing the TEIE bit to 0. TEI can also be canceled when data is written to SCTDR1 through the A-DMAC.                                |
|     |          |                  |     | 0: Transmit end interrupt request (TEI) is disabled   |
|     |          |                  |     | 1: Transmit end interrupt request (TEI) is enabled  |
|     | •        | •                |     |   |

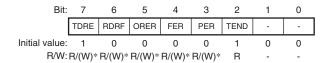
| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 1   | CKE1     | 0                | R/W | Clock Enable 1  |
|     |          |                  |     | Selects the SCI clock source and specifies the SCK pin functions in SCI clock synchronous mode.   |
|     |          |                  |     | The C/A bit in SCSMR1 should be set to 1 before setting the CKE1 bit. For details on clock source selection, refer to table 16.10 in section 16.4, Operation. |
|     |          |                  |     | 0: Internal clock, SCK pin used for synchronous clock output  |
|     |          |                  |     | 1: External clock, SCK pin used for clock output*   |
|     |          |                  |     | Note: * Do not set the CKE1 bit to 1 in asynchronous mode ( $C/\overline{A} = 0$ ). Otherwise, correct operation cannot be guaranteed.                        |
| 0   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.   |

## 16.3.7 Serial Status Register (SCSSR1)

SCSSR1 is an 8-bit register that contains status flags to indicate the SCI operating state.

The CPU can always read and write to SCSSR1, but cannot write 1 to status flags TDRE, RDRF, ORER, PER, and FER. These flags can be cleared to 0 only after 1 is read from the flags. The TEND flag is a read-only bit and cannot be modified.

SCSSR1 is initialized to H'84 by a power on reset or in standby mode.



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

| Bit | Bit Name | Initial<br>Value | R/W    | Description  |
|-----|----------|------------------|--------|--|
| 7   | TDRE     | 1                | R/(W)* | Transmit Data Register Empty   |
|     |          |                  |        | Indicates whether data has been transferred from the transmit data register (SCTDR1) to the transmit shift register (SCTSR1) and SCTDR1 has become ready to be written with next serial transmit data. |
|     |          |                  |        | 0: Indicates that SCTDR1 holds valid transmit data   |
|     |          |                  |        | [Clearing conditions]  |
|     |          |                  |        | • When 0 is written to TDRE after reading TDRE = 1   |
|     |          |                  |        | <ul> <li>When data is written to SCTDR1 through<br/>the A-DMAC</li> </ul>  |
|     |          |                  |        | 1: Indicates that SCTDR1 does not hold valid transmit data   |
|     |          |                  |        | [Setting conditions]   |
|     |          |                  |        | By a power-on reset  |
|     |          |                  |        | When the TE bit in SCSCR1 is 0   |
|     |          |                  |        | When data is transferred from SCTDR1 to<br>SCTSR1 and data can be written to SCTDR1  |

Apr 01, 2014

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |
|-----|----------|------------------|--------|---|
| 6   | RDRF     | 0                | R/(W)* | Receive Data Register Full  |
|     |          |                  |        | Indicates that the received data is stored in the receive data register (SCRDR1).   |
|     |          |                  |        | Indicates that valid received data is not stored in<br>SCRDR1   |
|     |          |                  |        | [Clearing conditions]   |
|     |          |                  |        | By a power-on reset   |
|     |          |                  |        | <ul> <li>When 0 is written to RDRF after reading RDRF =</li> <li>1</li> </ul>   |
|     |          |                  |        | <ul> <li>When data is read from SCRDR1 through<br/>the A-DMAC</li> </ul>  |
|     |          |                  |        | 1: Indicates that valid received data is stored in SCRDR1   |
|     |          |                  |        | [Setting condition]   |
|     |          |                  |        | <ul> <li>When serial reception ends normally and receive<br/>data is transferred from SCRSR1 to SCRDR1</li> </ul>   |
|     |          |                  |        | Note: SCRDR1 and the RDRF flag are not affected and retain their previous states even if an error is detected during data reception or if the RE bit in the serial control register (SCSCR1) is cleared to 0. If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the received data will be lost. |

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |  |  |
|-----|----------|------------------|--------|---|--|--|
| 5   | ORER     | 0                | R/(W)* | Overrun Error   |  |  |
|     |          |                  |        | Indicates that an overrun error occurred during reception, causing abnormal termination.  |  |  |
|     |          |                  |        | 0: Indicates that reception is in progress or was completed successfully*1  |  |  |
|     |          |                  |        | [Clearing conditions]   |  |  |
|     |          |                  |        | By a power-on reset   |  |  |
|     |          |                  |        | • When 0 is written to ORER after reading ORER = 1  |  |  |
|     |          |                  |        | Indicates that an overrun error occurred during reception* <sup>2</sup>   |  |  |
|     |          |                  |        | [Setting condition]   |  |  |
|     |          |                  |        | • When the next serial reception is completed while<br>RDRF = 1   |  |  |
|     |          |                  |        | Notes: 1. The ORER flag is not affected and retains its previous value when the RE bit in SCSCR1 is cleared to 0.   |  |  |
|     |          |                  |        | <ol> <li>The receive data prior to the overrun error<br/>is retained in SCRDR1, and the data<br/>received subsequently is lost. Subsequent<br/>serial reception cannot be continued while<br/>the ORER flag is set to 1.</li> </ol> |  |  |

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |  |  |  |
|-----|----------|------------------|--------|---|--|--|--|
| 4   | FER      | 0                | R/(W)* | Framing Error   |  |  |  |
|     |          |                  |        | Indicates that a framing error occurred during data reception in asynchronous mode, causing abnormal termination.   |  |  |  |
|     |          |                  |        | 0: Indicates that reception is in progress or was completed successfully*1  |  |  |  |
|     |          |                  |        | [Clearing conditions]   |  |  |  |
|     |          |                  |        | By a power-on reset   |  |  |  |
|     |          |                  |        | • When 0 is written to FER after reading FER = 1  |  |  |  |
|     |          |                  |        | Indicates that a framing error occurred during reception  |  |  |  |
|     |          |                  |        | [Setting condition]   |  |  |  |
|     |          |                  |        | When the SCI founds that the stop bit at the end  |  |  |  |
|     |          |                  |        | of the received data is 0 after completing reception* <sup>2</sup>  |  |  |  |
|     |          |                  |        | Notes: 1. The FER flag is not affected and retains its previous value when the RE bit in SCSCR1 is cleared to 0.  |  |  |  |
|     |          |                  |        | <ol> <li>In 2-stop-bit mode, only the first stop bit is<br/>checked for a value to 1; the second stop<br/>bit is not checked. If a framing error<br/>occurs, the receive data is transferred to<br/>SCRDR1 but the RDRF flag is not set.<br/>Subsequent serial reception cannot be<br/>continued while the FER flag is set to 1.</li> </ol> |  |  |  |

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |  |  |  |
|-----|----------|------------------|--------|---|--|--|--|
| 3   | PER      | 0                | R/(W)* | Parity Error  |  |  |  |
|     |          |                  |        | Indicates that a parity error occurred during data reception in asynchronous mode, causing abnormal termination.  |  |  |  |
|     |          |                  |        | 0: Indicates that reception is in progress or was completed successfully*1  |  |  |  |
|     |          |                  |        | <ul><li>[Clearing conditions]</li><li>By a power-on reset</li><li>When 0 is written to PER after reading PER = 1</li></ul>  |  |  |  |
|     |          |                  |        | <ul> <li>By a power-on reset</li> <li>When 0 is written to PER after reading PER = 1</li> <li>1: Indicates that a parity error occurred during</li> </ul>   |  |  |  |
|     |          |                  |        | • When 0 is written to PER after reading PER = 1  |  |  |  |
|     |          |                  |        | 1: Indicates that a parity error occurred during reception* <sup>2</sup>  |  |  |  |
|     |          |                  |        | [Setting condition]   |  |  |  |
|     |          |                  |        | When the number of 1s in the received data and  |  |  |  |
|     |          |                  |        | parity does not match the even or odd parity  |  |  |  |
|     |          |                  |        | specified by the $O/\overline{E}$ bit in the serial mode register (SCSMR1).   |  |  |  |
|     |          |                  |        | Notes: 1. The PER flag is not affected and retains its previous value when the RE bit in SCSCR1 is cleared to 0.  |  |  |  |
|     |          |                  |        | <ol> <li>If a parity error occurs, the receive data is<br/>transferred to SCRDR1 but the RDRF flag<br/>is not set. Subsequent serial reception<br/>cannot be continued while the PER flag is<br/>set to 1.</li> </ol> |  |  |  |

| ta was in SCTDR1 during it of the transmit character                            |
|---|
| · ·   |
| ded.  |
| nly and cannot be modified.   |
| sion is in progress   |
|   |
| TDRE after reading TDRE = 1   |
| sion has ended  |
|   |
|   |
| CSCR1 is 0  |
| ng transmission of the last bit smit character                                  |
| ND flag as a transmit end<br>MAC writes data to SCTDR1<br>XI interrupt request. |
|   |
| ad as 0. The write value  |
| r :   |

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

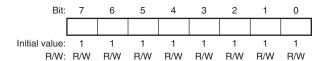
### 16.3.8 Bit Rate Register (SCBRR1)

SCBRR1 is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR1), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR1.

SCBRR1 is initialized to H'FF by a power on reset or in standby mode. The maximum bit rate in the asynchronous mode is 1.25 Mbps. Do not set more than the maximum in SCBRR1. Otherwise, normal operation cannot be guaranteed.

The SCBRR1 setting is calculated as follows:



Asynchronous mode:

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$

• Clock synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)

N: SCBRR1 setting for baud rate generator ( $0 \le N \le 255$ ) (The setting value should satisfy the electrical characteristics.)

Pφ: Operating frequency for peripheral modules (MHz)

n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 16.3.)

Table 16.3 SCSMR1 Settings

| CCCMD.  | 1 Settinas |
|---------|------------|
| SUSIVIR | ı sellinus |

|   |              |      | <del>-</del> |  |
|---|--------------|------|--------------|--|
| n | Clock Source | CKS1 | CKS0         |  |
| 0 | Рф           | 0    | 0            |  |
| 1 | Рф/4         | 0    | 1            |  |
| 2 | Ρφ/16        | 1    | 0            |  |
| 3 | Ρφ/64        | 1    | 1            |  |

The bit rate error in asynchronous is given by the following formula:

Error (%) = 
$$\left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 16.4 shows examples of SCBRR1 settings in asynchronous mode, and table 16.5 shows examples of SCBRR1 settings in clock synchronous mode.

Table 16.4 Bit Rates and SCBRR1 Settings in Asynchronous Mode

|          |   |     |           | - |     |           |
|----------|---|-----|-----------|---|-----|-----------|
| Bit Rate |   | 16  |           |   | 18  |           |
| (bits/s) | n | N   | Error (%) | n | N   | Error (%) |
| 110      | 3 | 141 | 0.03      | 3 | 159 | -0.12     |
| 150      | 3 | 103 | 0.16      | 3 | 116 | 0.16      |
| 300      | 3 | 51  | 0.16      | 2 | 233 | 0.16      |
| 600      | 3 | 25  | 0.16      | 2 | 116 | 0.16      |
| 1200     | 3 | 12  | 0.16      | 1 | 233 | 0.16      |
| 2400     | 2 | 25  | 0.16      | 1 | 116 | 0.16      |
| 4800     | 2 | 12  | 0.16      | 0 | 233 | 0.16      |
| 9600     | 1 | 25  | 0.16      | 0 | 116 | 0.16      |
| 19200    | 1 | 12  | 0.16      | 0 | 58  | -0.69     |
| 31250    | 1 | 7   | 0         | 0 | 35  | 0         |
| 38400    | 0 | 25  | 0.16      | 0 | 28  | 1.02      |

## Pφ (MHz)

| Bit Rate |   | 20  |           |   | 32  |           |
|----------|---|-----|-----------|---|-----|-----------|
| (bits/s) | n | N   | Error (%) | n | N   | Error (%) |
| 110      | 3 | 177 | -0.25     | _ | _   | _         |
| 150      | 3 | 129 | 0.16      | 3 | 207 | 0.16      |
| 300      | 3 | 64  | 0.16      | 3 | 103 | 0.16      |
| 600      | 2 | 129 | 0.16      | 2 | 207 | 0.16      |
| 1200     | 2 | 64  | 0.16      | 2 | 103 | 0.16      |
| 2400     | 1 | 129 | 0.16      | 1 | 207 | 0.16      |
| 4800     | 1 | 64  | 0.16      | 1 | 103 | 0.16      |
| 9600     | 0 | 129 | 0.16      | 0 | 207 | 0.16      |
| 19200    | 0 | 64  | 0.16      | 0 | 103 | 0.16      |
| 31250    | 0 | 39  | 0         | 0 | 63  | 0         |
| 38400    | 0 | 32  | -1.36     | 0 | 51  | 0.16      |

## Pφ (MHz)

|          |   |     |           | • • |     |           |  |
|----------|---|-----|-----------|-----|-----|-----------|--|
| Bit Rate |   | 36  |           |     | 40  |           |  |
| (bits/s) | n | N   | Error (%) | n   | N   | Error (%) |  |
| 110      | _ |     |           |     |     |           |  |
| 150      | 3 | 233 | 0.16      |     |     |           |  |
| 300      | 3 | 116 | 0.16      | 3   | 129 | 0.16      |  |
| 600      | 2 | 233 | 0.16      | 3   | 64  | 0.16      |  |
| 1200     | 2 | 116 | 0.16      | 2   | 129 | 0.16      |  |
| 2400     | 1 | 233 | 0.16      | 2   | 64  | 0.16      |  |
| 4800     | 1 | 116 | 0.16      | 1   | 129 | 0.16      |  |
| 9600     | 0 | 233 | 0.16      | 1   | 64  | 0.16      |  |
| 19200    | 0 | 116 | 0.16      | 0   | 129 | 0.16      |  |
| 31250    | 0 | 71  | 0         | 0   | 79  | 0         |  |
| 38400    | 0 | 58  | -0.68     | 0   | 64  | 0.16      |  |
|          | · | •   |           | ·   | •   |           |  |

Table 16.5 Bit Rates and SCBRR1 Settings in Clock Synchronous Mode

Pφ (MHz)

| Bit Rate |   | 16  |   | 18  |   | 20  |   | 32  |
|----------|---|-----|---|-----|---|-----|---|-----|
| (bits/s) | n | N   | n | N   | n | N   | n | N   |
| 10       | _ |     |   |     | _ |     | _ | _   |
| 250      | 3 | 249 | _ |     | _ |     |   | _   |
| 500      | 3 | 124 | _ |     | _ | _   | 3 | 249 |
| 1k       | 2 | 249 |   |     | _ |     | 3 | 124 |
| 2.5k     | 2 | 99  |   | _   | 2 | 124 | 2 | 199 |
| 5k       | 1 | 199 | 1 | 224 | 1 | 249 | 2 | 99  |
| 10k      | 1 | 99  | _ |     | _ |     | 1 | 199 |
| 25k      | 0 | 159 | 1 | 44  | 1 | 49  | 1 | 79  |
| 50k      | 0 | 79  | 0 | 89  | 1 | 24  | 0 | 159 |
| 100k     | 0 | 39  | 0 | 44  | 0 | 49  | 0 | 79  |
| 250k     | 0 | 15  | 0 | 17  | 0 | 19  | 0 | 31  |
| 500k     | 0 | 7   | 0 | 8   | 0 | 9   | 0 | 15  |
| 1M       | 0 | 3   |   |     | 0 | 4   | 0 | 7   |
| 2.5M     | _ |     |   | _   | 0 | 1   |   | _   |

|                   |   |     | Pφ (MHz) |     |  |
|-------------------|---|-----|----------|-----|--|
|                   |   | 36  |          | 40  |  |
| Bit Rate (bits/s) | n | N   | n        | N   |  |
| 10                | _ |     |          | _   |  |
| 250               | _ |     |          | _   |  |
| 500               | _ |     |          | _   |  |
| 1k                | _ |     |          |     |  |
| 2.5k              | 2 | 224 | 2        | 249 |  |
| 5k                | _ |     | 2        | 124 |  |
| 10k               | 1 | 224 | 1        | 249 |  |
| 25k               | 1 | 89  | 1        | 99  |  |
| 50k               | 0 | 179 | 1        | 49  |  |
| 100k              | 0 | 89  | 1        | 24  |  |
| 250k              | 0 | 35  | 0        | 39  |  |
| 500k              | 0 | 17  | 0        | 19  |  |
| 1M                | 0 | 8   | 0        | 9   |  |
| 2.5M              | _ |     | 0        | 3   |  |

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

[Legend]

Blank: Setting is not possible.

—: Setting possible, but error will result.

Table 16.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 16.7 and 16.8 list the maximum rates for external clock input.

Table 16.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

|          |                           | Settings |   |  |  |  |  |
|----------|---------------------------|----------|---|--|--|--|--|
| Pφ (MHz) | Maximum Bit Rate (bits/s) | n        | N |  |  |  |  |
| 16       | 1000000                   | 0        | 0 |  |  |  |  |
| 18       | 1125000                   | 0        | 0 |  |  |  |  |
| 20       | 1250000                   | 0        | 0 |  |  |  |  |
| 32       | 1000000                   | 0        | 1 |  |  |  |  |
| 36       | 1125000                   | 0        | 1 |  |  |  |  |
| 40       | 1250000                   | 0        | 1 |  |  |  |  |

Table 16.7 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode:  $t_{scyc}$ =  $12_{cyc}$ )

| Pφ (MHz) | External Input Clock (MHz) | Maximum Bit Rate (bits/s) |  |  |  |  |  |
|----------|----------------------------|---------------------------|--|--|--|--|--|
| 16       | 1.33                       | 1333333                   |  |  |  |  |  |
| 18       | 1.5                        | 1500000                   |  |  |  |  |  |
| 20       | 1.67                       | 1666666                   |  |  |  |  |  |

Table 16.8 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode:  $t_{scyc} = 16_{cvc}$ )

| Pφ (MHz) | External Input Clock (MHz) | Maximum Bit Rate (bits/s) |  |  |  |  |  |
|----------|----------------------------|---------------------------|--|--|--|--|--|
| 32       | 2                          | 2000000                   |  |  |  |  |  |
| 36       | 2.25                       | 2250000                   |  |  |  |  |  |
| 40       | 2.5                        | 2500000                   |  |  |  |  |  |

## 16.4 Operation

#### **16.4.1** Overview

For serial communications, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses.

Asynchronous or clock synchronous mode is selected and the transmit format is specified in the serial mode register (SCSMR1) as shown in table 16.9. The SCI clock source is selected by the combination of the  $C/\overline{A}$  bit in SCSMR1 and the CKE1 bit in the serial control register (SCSCR1) as shown in table 16.10.

## (1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits.
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and breaks.
- SCI clock source: The SCI operates on the clock supplied by the on-chip baud rate generator.

## (2) Clock Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Table 16.9 SCSMR1 Settings and SCI Communication Formats

| SCSMR1 Settings |              |             | าgs           |                      | SCI Communication Format |            |                    |  |  |  |  |
|-----------------|--------------|-------------|---------------|----------------------|--------------------------|------------|--------------------|--|--|--|--|
| Bit 7<br>C/A    | Bit 6<br>CHR | Bit 5<br>PE | Bit 3<br>STOP | Mode                 | Data Length              | Parity Bit | Stop Bit<br>Length |  |  |  |  |
| 0               | 0            | 0           | 0             | Asynchronous         | 8-bit                    | Not set    | 1 bit              |  |  |  |  |
|                 |              |             | 1             | _                    |                          |            | 2 bits             |  |  |  |  |
|                 |              | 1           | 0             | _                    |                          | Set        | 1 bit              |  |  |  |  |
|                 |              |             | 1             | _                    |                          |            | 2 bits             |  |  |  |  |
|                 | 1            | 0           | 0             | _                    | 7-bit                    | Not set    | 1 bit              |  |  |  |  |
|                 |              |             | 1             | _                    |                          |            | 2 bits             |  |  |  |  |
|                 |              | 1           | 0             | _                    |                          | Set        | 1 bit              |  |  |  |  |
|                 |              |             | 1             | _                    |                          |            | 2 bits             |  |  |  |  |
| 1               | х            | х           | Х             | Clock<br>synchronous | 8-bit                    | Not set    | None               |  |  |  |  |

[Legend]

x: Don't care

Table 16.10 SCSMR1 and SCSCR1 Settings and SCI Clock Source Selection

| SCSMR1<br>Setting | SCSCR1<br>Setting |                   |              |                               |
|-------------------|-------------------|-------------------|--------------|-------------------------------|
| Bit 7<br>C/Ā      | Bit 1<br>CKE1     | <br>Mode          | Clock Source | SCK Pin Function              |
| 0                 | 0*                | Asynchronous      | Internal     | SCI does not use the SCK pin. |
| 1                 | 0                 | Clock synchronous | Internal     | Serial clock is output.       |
|                   | 1                 | <del></del>       | External     | Input the serial clock.       |

Note: \* Do not set the CKE1 bit to 1 in asynchronous mode. Otherwise, correct operation cannot be guaranteed.

## 16.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communications are synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communications are possible. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 16.2 shows the general format of asynchronous serial communications. In asynchronous serial communications, the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communications when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

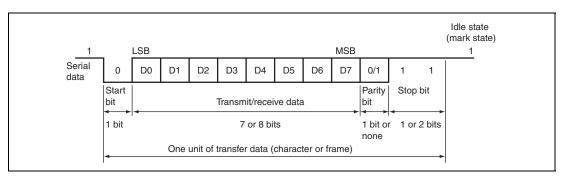


Figure 16.2 Example of Data Format in Asynchronous Communications (8-Bit Data with Parity and Two Stop Bits)

#### **Transmit/Receive Formats (1)**

Table 16.11 shows the transfer formats that can be selected in asynchronous mode. Any of eight transfer formats can be selected according to the SCSMR1 settings.

**Table 16.11 Serial Transfer Formats (Asynchronous Mode)** 

| SCSMR1 Settings |    |      |   |                      | Serial Transfer Format and Frame Length |   |   |   |   |   |      |      |           |      |  |
|-----------------|----|------|---|----------------------|---|---|---|---|---|---|------|------|-----------|------|--|
| CHR             | PE | STOP | 1 | 2                    | 3                                       | 4 | 5 | 6 | 7 | 8 | 9    | 10   | 11        | 12   |  |
| 0               | 0  | 0    | S |                      | 8-bit data                              |   |   |   |   |   |      |      | -         |      |  |
| 0               | 0  | 1    | S |                      | 8-bit data                              |   |   |   |   |   |      | STOP | STOP STOP |      |  |
| 0               | 1  | 0    | S |                      | 8-bit data                              |   |   |   |   |   |      |      | STOP      |      |  |
| 0               | 1  | 1    | S |                      | 8-bit data                              |   |   |   |   |   |      | Р    | STOP      | STOP |  |
| 1               | 0  | 0    | S |                      | 7-bit data STOP                         |   |   |   |   |   |      |      |           |      |  |
| 1               | 0  | 1    | S |                      | 7-bit data STOP                         |   |   |   |   |   | STOP | -    |           |      |  |
| 1               | 1  | 0    | S |                      | 7-bit data P                            |   |   |   |   |   |      | STOP | -         |      |  |
| 1               | 1  | 1    | s | 7-bit data P STOP ST |   |   |   |   |   |   | STOP |      |           |      |  |

[Legend]

Start bit STOP: Stop bit Parity bit

#### **(2)** Clock

An internal clock generated by the on-chip baud rate generator can be used as the SCI transmit/receive clock. The CKE1 bit in the serial control register (SCSCR1) must be set to 0 in asynchronous mode. Otherwise, correct operation cannot be guaranteed. For clock source selection, refer to table 16.10.

## (3) Transmitting and Receiving Data

### **SCI Initialization (Asynchronous Mode):**

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR1), then initialize the SCI as follows.

If the TIE bit in SCSCR1 is set to 1 when the TE bit is 0, a transmit-data-empty interrupt (TXI) request is generated. To disable the TXI request when initializing the SCI, the TE and TIE bits must be cleared to 0 simultaneously. If the RIE bit in SCSCR1 and the RDRF flag are set to 1 when the RE bit is 0, a receive-data-full interrupt (RXI) request is generated. To disable the RXI request when initializing the SCI, the RE and RIE bits must be cleared to 0 simultaneously.

When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing the TE bit to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR1). Clearing the RE bit to 0, however, does not initialize the RDRF, PER, FER, and ORER flags or receive data register (SCRDR1), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

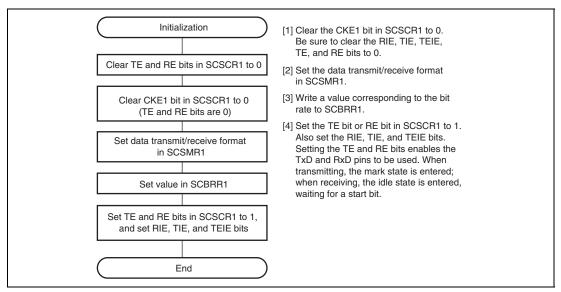


Figure 16.3 Sample Flowchart for SCI Initialization

### **Transmitting Serial Data (Asynchronous Mode):**

Figure 16.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCI for transmission.

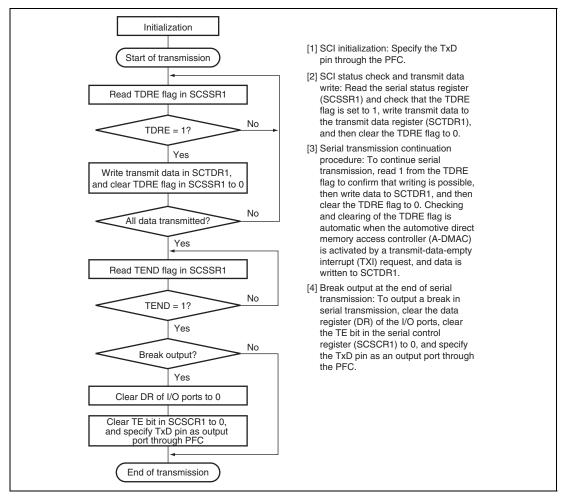


Figure 16.4 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCI operates as described below.

- The SCI monitors the TDRE flag in the serial status register (SCSSR1). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR1) and transfers the data from SCTDR1 to the transmit shift register (SCTSR1).
- 2. After transferring data from SCTDR1 to SCTSR1, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in the serial control register (SCSCR1) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated.

The serial transmit data is sent from the TxD pin in the following order.

- A. Start bit: One-bit 0 is output.
- B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
- C. Parity bit: One parity bit (even or odd parity) is output. (A format in which parity bit is not output can also be selected.)
- D. Stop bit(s): One or two 1 bits (stop bits) are output.
- E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
- 3. The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is 0, the data is transferred from SCTDR1 to SCTSR1, the stop bit is sent, and then serial transmission of the next frame is started.

If the TDRE flag is 1, the TEND flag in SCSSR1 is set to 1, the stop bit is sent, and then the "mark state" is entered in which 1 is output. If the TEIE bit in SCSCR1 is set to 1 at this time, a TEI interrupt request is generated.

Figure 16.5 shows an example of the operation for transmission.

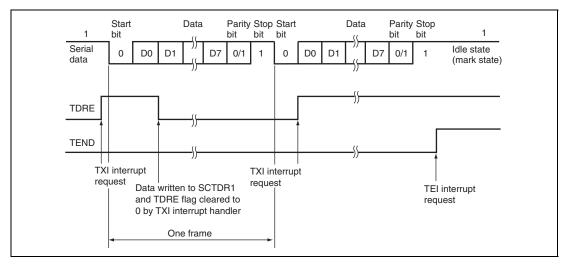


Figure 16.5 Example of Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

### **Receiving Serial Data (Asynchronous Mode):**

Figure 16.6 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCI for reception.

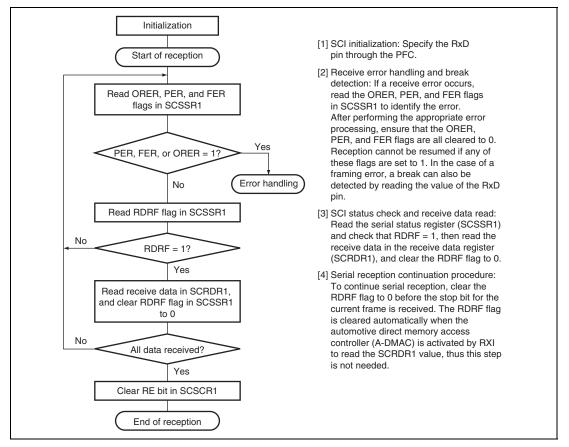


Figure 16.6 Sample Flowchart for Receiving Serial Data (1)

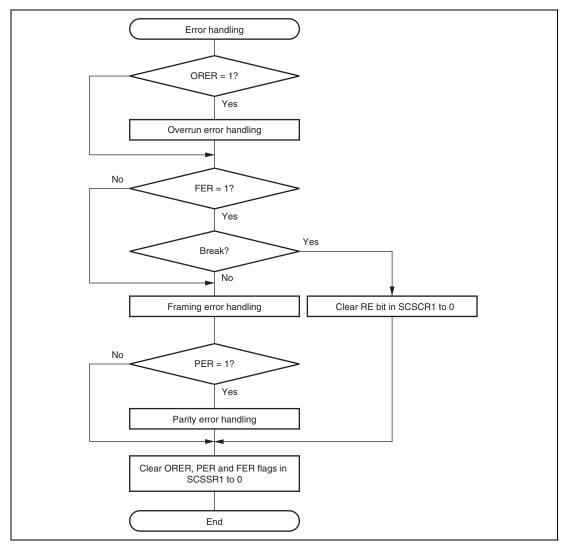


Figure 16.6 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCI operates as described below.

- 1. The SCI monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
- 2. The received data is stored in SCRSR1 in LSB-to-MSB order.
- 3. The parity bit and stop bit are received.

After receiving these bits, the SCI carries out the following checks.

- A. Parity check: The SCI counts the number of 1s in the received data and checks whether the count matches the even or odd parity specified by the O/E bit in the serial mode register (SCSMR1).
- B. Stop bit check: The SCI checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
- C. Status check: The SCI checks whether the RDRF flag is 0 and the received data can be transferred from the receive shift register (SCRSR1) to SCRDR1.

If all the above checks are passed, the RDRF flag is set to 1 and the received data is stored in SCRDR1. If a receive error is detected, the SCI operates as shown in table 16.12.

Note: When a receive error occurs, subsequent reception cannot be continued. In addition, the RDRF flag will not be set to 1 after reception; be sure to clear the error flag to 0.

4. If the RIE bit in SCSCR1 is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated. If the RIE bit in SCSCR1 is set to 1 when the ORER, PER, or FER flag changes to 1, a receive error interrupt (ERI) request is generated.

Table 16.12 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 SCSSR1 is set to 1    | The received data is not transferred from SCRSR1 to SCRDR1. |
| Framing error | FER          | When the stop bit is 0   | The received data is transferred from SCRSR1 to SCRDR1.     |
| Parity error  | PER          | When the received data does<br>not match the even or odd<br>parity specified in SCSMR1 | The received data is transferred from SCRSR1 to SCRDR1.     |



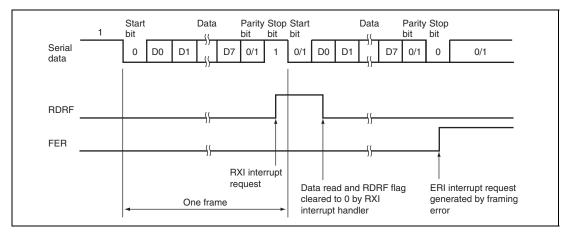


Figure 16.7 Example of SCI Receive Operation (8-Bit Data, Parity, One Stop Bit)

#### 16.4.3 Clock Synchronous Mode

In clock synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communications.

The SCI transmitter and receiver are independent, so full-duplex communications are possible while sharing the same clock. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 16.8 shows the general format in clock synchronous serial communications.

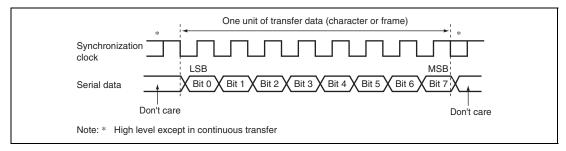


Figure 16.8 Data Format in Clock Synchronous Communications

In clock synchronous serial communications, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In clock synchronous mode, the SCI transmits or receives data by synchronizing with the rising edge of the serial clock.

#### (1) Communication Format

The data length is fixed at eight bits. No parity bit can be added.

#### (2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The SCI clock source is selected by the combination of the  $C/\overline{A}$  bit in SCSMR1 and the CKE1 bit in the serial control register (SCSCR1). For clock source selection, refer to table 16.10.

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

### (3) Transmitting and Receiving Data

### **SCI Initialization (Clock Synchronous Mode):**

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR1), then initialize the SCI as follows.

If the TIE bit in SCSCR1 is set to 1 when the TE bit is 0, a transmit-data-empty interrupt (TXI) request is generated. To disable the TXI request when initializing the SCI, the TE and TIE bits must be cleared to 0 simultaneously. If the RIE bit in SCSCR1 and the RDRF flag are set to 1 when the RE bit is 0, a receive-data-full interrupt (RXI) request is generated. To disable the RXI request when initializing the SCI, the RE and RIE bits must be cleared to 0 simultaneously.

When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR1). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (SCRDR1), which retain their previous contents.

Figure 16.9 shows a sample flowchart for initializing the SCI.

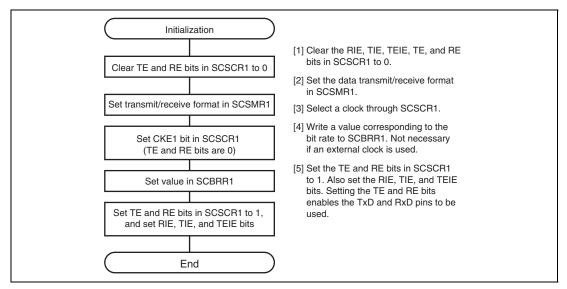


Figure 16.9 Sample Flowchart for SCI Initialization

Page 730 of 1824

#### Transmitting Serial Data (Clock Synchronous Mode):

Figure 16.10 shows a sample flowchart for transmitting serial data.

Use the following procedure for serial data transmission after enabling the SCI for transmission.

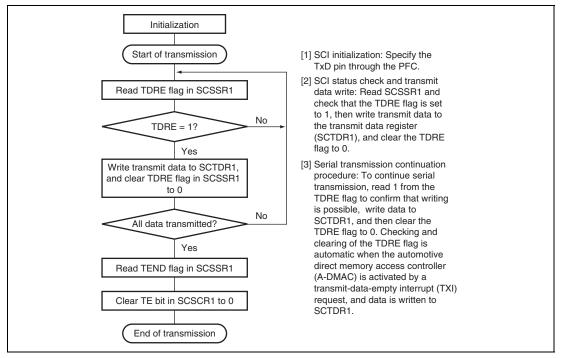


Figure 16.10 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

- 1. The SCI monitors the TDRE flag in the serial status register (SCSSR1). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR1) and transfers the data from SCTDR1 to the transmit shift register (SCTSR1).
- 2. After transferring data from SCTDR1 to SCTSR1, the SCI sets the TDRE flag to 1 and starts transmission. If the transmit-data-empty interrupt enable bit (TIE) in the serial control register (SCSCR1) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated. If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data is output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).
- 3. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7). If the TDRE flag is 0, the data is transferred from SCTDR1 to SCTSR1 and serial transmission of the next frame is started, If the TDRE flag is 1, the TEND flag in SCSSR1 is set to 1, the MSB (bit 7) is sent, and then the TxD pin holds the states.
  - If the TEIE bit in SCSCR1 is set to 1 at this time, a TEI interrupt request is generated.
- 4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 16.11 shows an example of SCI transmit operation.

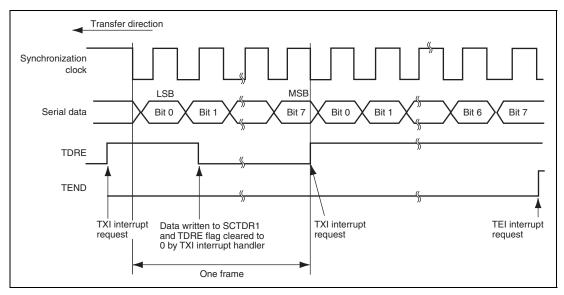


Figure 16.11 Example of SCI Transmit Operation

#### **Receiving Serial Data (Clock Synchronous Mode):**

Figure 16.12 shows a sample flowchart for receiving serial data. Use the following procedure for serial data reception after enabling the SCI for reception.

When switching from asynchronous mode to clock synchronous mode, make sure that the ORER, PER, and FER flags are all cleared to 0. If the FER or PER flag is set to 1, the RDRF flag will not be set and data reception cannot be started.

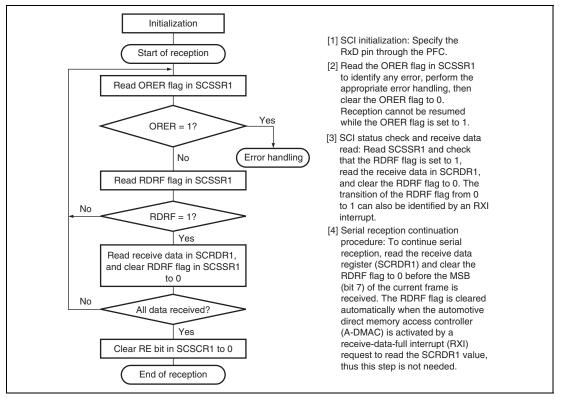


Figure 16.12 Sample Flowchart for Receiving Serial Data (1)

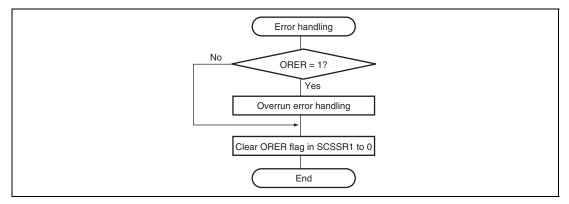


Figure 16.12 Sample Flowchart for Receiving Serial Data (2)

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 SCRSR1 in order from the LSB to the MSB. After receiving the data, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from SCRSR1 to SCRDR1. If this check is passed, the SCI sets the RDRF flag to 1 and stores the received data in SCRDR1. If a receive error is detected, the SCI operates as shown in table 16.11. In this state, subsequent reception cannot be continued. In addition, as the RDRF flag will be set to 1 after reception, be sure to clear the RDRF flag to 0.
- 3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCSCR1, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the RIE bit in SCSCR1 is also set to 1, the SCI requests a receive error interrupt (ERI).

Figure 16.13 shows an example of SCI receive operation.

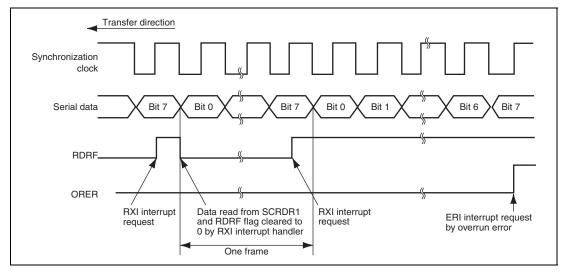


Figure 16.13 Example of SCI Receive Operation

### Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode):

Figure 16.14 shows a sample flowchart for transmitting and receiving serial data simultaneously.

Use the following procedure for serial data transmission and reception after enabling the SCI for transmission and reception.

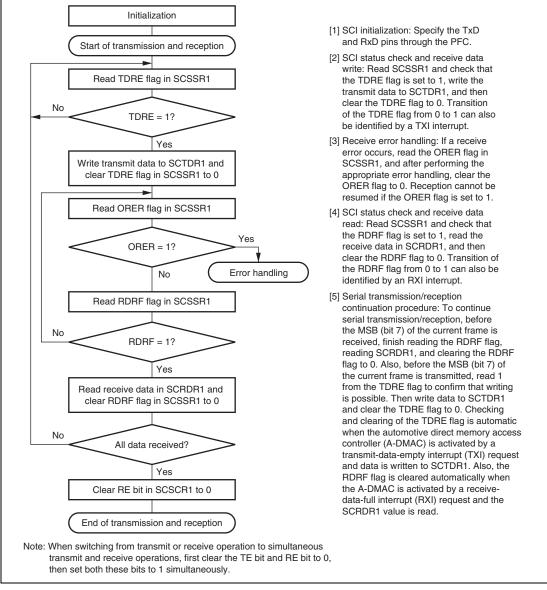


Figure 16.14 Sample Flowchart for Transmitting/Receiving Serial Data

# 16.5 SCI Interrupt Sources and A-DMAC

The SCI has four interrupt sources: transmit end (TEI), receive error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI) interrupt requests. When DMA transfer to the A-DMAC is not enabled, these interrupt requests are sent to the interrupt controller separately. When DMA transfer is enabled, RXI and TXI requests are handled as DMA requests.

Table 16.13 shows the interrupt sources. The interrupt sources are enabled or disabled by means of the TIE, RIE, and TEIE bits in SCSCR1. DMA requests are enabled or disabled by the TIE and RIE bits in SCSCR1.

If the TDRE flag in the serial status register (SCSSR1) is set to 1 when the TIE bit in SCSCR1 is 1, a transmit-data-empty interrupt (TXI) request is generated. This request can be used to activate the automotive direct memory access controller (A-DMAC) to transfer data. To disable the TXI request to the A-DMAC, clear the TIE bit to 0.

The TDRE flag is automatically cleared to 0 when data is written to the transmit data register (SCTDR1) through the A-DMAC.

If the RDRF flag in SCSSR1 is set to 1 when the RIE bit in SCSCR1 is 1, a receive-data-full interrupt (RXI) request is generated. This request can be used to activate the A-DMAC to transfer data. To disable the RXI request to the A-DMAC, clear the RIE bit to 0.

The RDRF flag is automatically cleared to 0 when data is read from the receive data register (SCRDR1) through the A-DMAC.

When the ORER, FER, or PER flag in SCSSR1 is set to 1, an ERI interrupt request is generated. This request cannot be used to activate the A-DMAC. When processing the received data through the A-DMAC and handling the receive error by an interrupt requested to the CPU, set the RIE bit to 1.

When the TEND flag in SCSSR1 is set to 1, a TEI interrupt request is generated. This request cannot be used to activate the A-DMAC.

The TXI interrupt indicates that transmit data can be written, and the TEI interrupt indicates that transmission has been completed.

**Table 16.13 SCI Interrupt Sources** 

| Interrupt<br>Source | Description   | A-DMAC Activation | Priority on Reset<br>Release |
|---------------------|---|-------------------|------------------------------|
| ERI                 | Interrupt caused by receive error (ORER, FER, or PER) | Not possible      | High<br>♠                    |
| 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      | Low                          |

# 16.6 Usage Notes

### 16.6.1 SCTDR1 Writing and TDRE Flag

The TDRE flag in the serial status register (SCSSR1) is a status flag indicating transferring of transmit data from SCTDR1 into SCTSR1. The SCI sets the TDRE flag to 1 when it transfers data from SCTDR1 to SCTSR1.

Data can be written to SCTDR1 regardless of the TDRE bit status.

If new data is written in SCTDR1 when TDRE is 0, however, the old data stored in SCTDR1 will be lost because the data has not yet been transferred to SCTSR1. Before writing transmit data to SCTDR1, be sure to check that the TDRE flag is set to 1.

# 16.6.2 Multiple Receive Error Occurrence

If multiple receive errors occur at the same time, the status flags in SCSSR1 are set as shown in table 16.14. When an overrun error occurs, data is not transferred from the receive shift register (SCRSR1) to the receive data register (SCRDR1) and the received data will be lost.

Table 16.14 SCSSR1 Status Flag Values and Transfer of Received Data

|  | S    | CSSR1 S | Receive Data Transfer from SCRSR1 to |     |                 |
|--|------|---------|--------------------------------------|-----|-----------------|
| Receive Errors Generated                     | RDRF | ORER    | FER                                  | PER | SCRDR1          |
| Overrun error                                | 1    | 1       | 0                                    | 0   | Not transferred |
| Framing error                                | 0    | 0       | 1                                    | 0   | Transferred     |
| Parity error                                 | 0    | 0       | 0                                    | 1   | Transferred     |
| Overrun error + framing error                | 1    | 1       | 1                                    | 0   | Not transferred |
| Overrun error + parity error                 | 1    | 1       | 0                                    | 1   | Not transferred |
| Framing error + parity error                 | 0    | 0       | 1                                    | 1   | Transferred     |
| Overrun error + framing error + parity error | 1    | 1       | 1                                    | 1   | Not transferred |

### 16.6.3 Break Detection and Processing

Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCRDR1 is halted in the break state, the SCI receiver continues to operate.

# 16.6.4 Sending a Break Signal

The TxD pin is used as a general I/O pin. The I/O condition and level of the TxD pin are determined by the data register (DR) of the I/O ports and the control register of the pin function controller (PFC). This feature can be used to send a break signal.

Until the TE bit is set to 1 (enabling transmission) after initialization of serial transmission, the TxD pin does not work. Until PFC settings complete, mark state is achieved using DR. Therefore, the TxD pin should be specified as an output port to output 1 initially. To send a break signal during serial transmission, clear DR to 0, then specify the TxD pin as an output port. When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state.

# 16.6.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCI operates on a base clock with a frequency of 16 times the transfer rate in asynchronous mode. In reception, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 16.15.

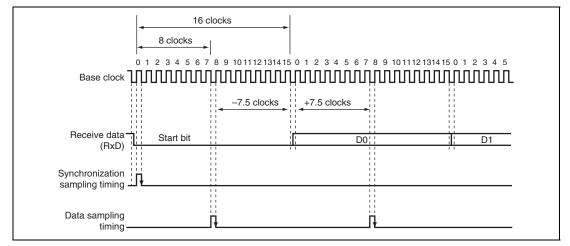


Figure 16.15 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

### **Equation 1:**

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1+F) \right| \times 100 \%$$

M: Receive margin (%)

N: Ratio of bit rate to clock (N = 16)

D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

# **Equation 2:**

When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

#### 16.6.6 Note on Using A-DMAC

When data is written to SCTDR1 by activating the A-DMAC by a TXI interrupt, the transmit end (TEND) flag value becomes undefined. In this case, do not use the TEND flag as the transmit end flag.

When the external clock source is used for the clock for synchronization, input the external clock after waiting for five or more  $P\phi$  cycles after SCTDR1 is modified through the A-DMAC. If a transmit clock is input within four cycles after SCTDR1 is modified, a malfunction may occur (figure 16.16).

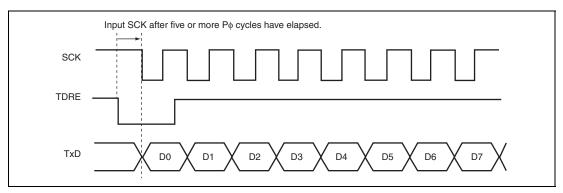


Figure 16.16 Example of Clock Synchronous Transfer Using A-DMAC

# 16.6.7 Note on Using External Clock in Clock Synchronous Mode

TE and RE must be set to 1 after waiting for four or more  $P\phi$  cycles after the SCK external clock is changed from 0 to 1.

TE and RE must be set to 1 only while the SCK external clock is 1.

# 16.6.8 Note on Using A-DMAC

When the A-DMAC is used for transmitting data, the TIE bit in SCSCR1 should be set to 1 to enable a transmit-data-empty interrupt (TXI) request. When the A-DMAC is used for receiving data, the RIE bit in SCSCR1 should be set to 1 to enable a receive-data-full interrupt (RXI) request.

#### 16.6.9 Serial Ports

Note that when the SCI pins are read through the serial ports, the value two  $P\phi$  cycles before is read.

# 16.6.10 Note on Reception Only, with SCK Output, in Clock Synchronous Mode

In clock synchronous mode, only in the communications configured as reception only, with SCK pin used for output, the maximum baud rate is 1.25 Mbps.

# Section 17 Renesas Serial Peripheral Interface (RSPI)

This LSI includes three-channel Renesas Serial Peripheral Interfaces (RSPI).

The RSPI has three channels which are independent of each other and is capable of full-duplex high-speed serial communications with multiple processors and peripheral devices.

#### 17.1 **Features**

The RSPI is a four-wire serial interface with the following configuration of signal lines.

RSPCK (RSPI clock)

SSL (slave select)

MOSI (master output/slave in)

MISO (master in/slave out)

- Capable of multi-master, single-master, and slave mode serial communications.
- Bit rate

In master mode: 10.0 MHz at maximum when  $P\phi = 20/40$  MHz

An internal baud rate generator generates RSPCK by dividing  $P\phi$  (division by 2 to 4096)

In slave mode: 2.5 MHz at maximum when  $P\phi 20/40 = MHz$ 

Uses an external input clock as the serial clock (divided by 8 when one-time multiplication has been set for the peripheral clock; division by 16 when twotimes multiplication has been set for the peripheral clock).

- Send/receive buffers in a double-buffer configuration.
- Data format

Switchable MSB first/LSB first.

Transfer bit length changeable to 8-16 bits.

- Modifiable RSPCK polarity/phase.
- SSL control function

Modifiable SSL0 to SSL7 polarity.

In single-master mode, outputs SSL0 to SSL7 signals.

In multi-master mode, SSL0 signal for input, and SSL1 to SSL7 signals for either output or Hi-**Z**.

In slave mode, SSL0 signal for input, and SSL1 to SSL7 signals for Hi-Z.

- In master mode, MOSI signal values can be set during SSL negation.
- Switchable CMOS output and open drain output.

- Serial transmission can be executed in sequential loops.
  - Loops comprised of a maximum of eight commands.

For each command, the following transfer formats can be set:

- SSL0 to SSL7 signal output value
- Data format
- Burst transfer
- A delay from SSL output assertion to RSPCK operation (RSPCK delay)
- A delay from RSPCK stop to SSL output negation (SSL negation delay)
- Wait for next-access SSL output (next-access delay)
- RSPCK polarity and phase
- Bit rate
- Receive buffer full flags provided (available as interrupts/DMA requests)
- Transmission buffer empty flags provided (available as interrupts/DMA requests)
- Mode default error flag provided (available as an interrupt request)
- Overrun error flag provided (available as an interrupt request)

Figure 17.1 shows an RSPI block diagram for one channel. When the CPU accesses the RSPI control registers, a peripheral bus (P-bus) is used. When A-DMAC accesses the RSPI data registers (SPDR), a special bus (RSPI bus) is used.

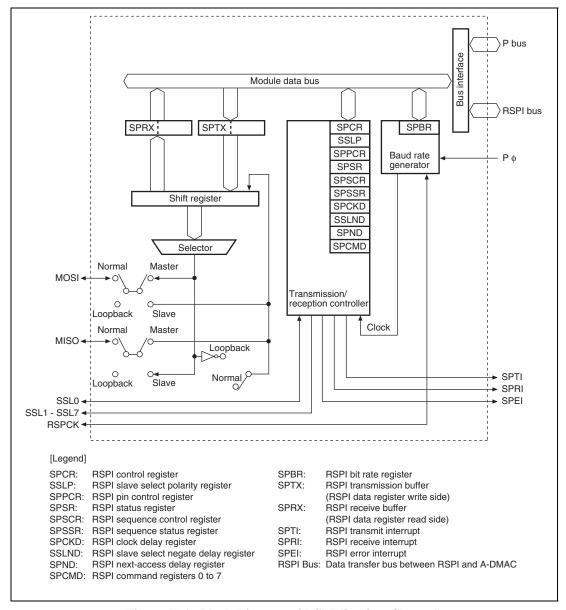


Figure 17.1 Block Diagram of RSPI (for One Channel)

# 17.2 Input/Output Pins

Table 17.1 shows the RSPI pin configuration. The RSPI automatically switches input/output directions of the pins. Pins SSLA0, SSLB0, and SSLC0 are set to outputs when the RSPI is a single master and are set to inputs when the RSPI is a multi master or a slave. Pins RSPCKA, MOSIA, MISOA, RSPCKB, MOSIB, MISOB, RSPCKC, MOSIC, and MISOC are set to inputs or outputs according to the following settings: whether the RSPI functions as a master or a slave and the input levels of pins SSLA0, SSLB0, and SSLC0 (see section 17.4.2, Controlling RSPI Pins).

**Table 17.1 Pin Configuration** 

| Channel | Pin Name                 | Symbol | I/O    | Function                    |
|---------|--------------------------|--------|--------|-----------------------------|
| A       | RSPI clock pin           | RSPCKA | I/O    | RSPI_A clock input/output   |
|         | Master transmit data pin | MOSIA  | I/O    | RSPI_A master transmit data |
|         | Slave transmit data pin  | MISOA  | I/O    | RSPI_A slave transmit data  |
|         | Slave select 0 pin       | SSLA0  | I/O    | RSPI_A slave select         |
|         | Slave select 1 pin       | SSLA1  | Output | RSPI_A slave select         |
|         | Slave select 2 pin       | SSLA2  | Output | RSPI_A slave select         |
|         | Slave select 3 pin       | SSLA3  | Output | RSPI_A slave select         |
|         | Slave select 4 pin       | SSLA4  | Output | RSPI_A slave select         |
|         | Slave select 5 pin       | SSLA5  | Output | RSPI_A slave select         |
|         | Slave select 6 pin       | SSLA6  | Output | RSPI_A slave select         |
|         | Slave select 7 pin       | SSLA7  | Output | RSPI_A slave select         |
| В       | RSPI clock pin           | RSPCKB | I/O    | RSPI_B clock input/output   |
|         | Master transmit data pin | MOSIB  | I/O    | RSPI_B master transmit data |
|         | Slave transmit data pin  | MISOB  | I/O    | RSPI_B slave transmit data  |
|         | Slave select 0 pin       | SSLB0  | I/O    | RSPI_B slave select         |
|         | Slave select 1 pin       | SSLB1  | Output | RSPI_B slave select         |
|         | Slave select 2 pin       | SSLB2  | Output | RSPI_B slave select         |
|         | Slave select 3 pin       | SSLB3  | Output | RSPI_B slave select         |

| Channel | Pin Name                 | Symbol | I/O    | Function                    |
|---------|--------------------------|--------|--------|-----------------------------|
| С       | RSPI clock pin           | RSPCKC | I/O    | RSPI_C clock input/output   |
|         | Master transmit data pin | MOSIC  | I/O    | RSPI_C master transmit data |
|         | Slave transmit data pin  | MISOC  | I/O    | RSPI_C slave transmit data  |
|         | Slave select 0 pin       | SSLC0  | I/O    | RSPI_C slave select         |
|         | Slave select 1 pin       | SSLC1  | Output | RSPI_C slave select         |
|         | Slave select 2 pin       | SSLC2  | Output | RSPI_C slave select         |
|         | Slave select 3 pin       | SSLC3  | Output | RSPI_C slave select         |

Note: Pin names RSPCK, MOSI, MISO, and SSL0 to SSL7 are used in the description for all channels, omitting the channel designation.

# 17.3 Register Descriptions

The RSPI has the registers shown in table 17.2. These registers enable the RSPI to perform the following controls: specifying master/slave modes, specifying a transfer format, and controlling the transmitter and receiver.

**Table 17.2 Register Configuration** 

| Chan-<br>nel | Register Name                               | Symbol* <sup>1</sup> | R/W     | Initial<br>Value | Address    | Access<br>Size |
|--------------|---|----------------------|---------|------------------|------------|----------------|
| A            | RSPI control register A                     | SPCRA                | R/W     | H'00             | H'FFFFB000 | 8, 16          |
|              | RSPI slave select polarity register A       | SSLPA                | R/W     | H'00             | H'FFFFB001 | 8              |
|              | RSPI pin control register A                 | SPPCRA               | R/W     | H'00             | H'FFFFB002 | 8, 16          |
|              | RSPI status register A                      | SPSRA                | R/(W)*2 | H'20             | H'FFFFB003 | 8              |
|              | RSPI data register A                        | SPDRA                | R/W     | H'0000           | H'FFFFB004 | 16             |
|              | RSPI sequence control register A            | SPSCRA               | R/W     | H'00             | H'FFFFB008 | 8, 16          |
|              | RSPI sequence status register A             | SPSSRA               | R       | H'00             | H'FFFFB009 | 8              |
|              | RSPI bit rate register A                    | SPBRA                | R/W     | H'FF             | H'FFFFB00A | 8              |
|              | RSPI clock delay register A                 | SPCKDA               | R/W     | H'00             | H'FFFFB00C | 8, 16          |
|              | RSPI slave select negation delay register A | SSLNDA               | R/W     | H'00             | H'FFFFB00D | 8              |
|              | RSPI next-access delay register A           | SPNDA                | R/W     | H'00             | H'FFFFB00E | 8              |
|              | RSPI command register A0                    | SPCMD0               | R/W     | H'070D           | H'FFFFB010 | 16             |
|              | RSPI command register A1                    | SPCMD1               | R/W     | H'070D           | H'FFFFB012 | 16             |
|              | RSPI command register A2                    | SPCMD2               | R/W     | H'070D           | H"FFFFB014 | 16             |
|              | RSPI command register A3                    | SPCMD3               | R/W     | H'070D           | H'FFFFB016 | 16             |
|              | RSPI command register A4                    | SPCMD4               | R/W     | H'070D           | H'FFFFB018 | 16             |
|              | RSPI command register A5                    | SPCMD5               | R/W     | H'070D           | H'FFFFB01A | 16             |
|              | RSPI command register A6                    | SPCMD6               | R/W     | H'070D           | H'FFFFB01C | 16             |
|              | RSPI command register A7                    | SPCMD7               | R/W     | H'070D           | H'FFFFB01E | 16             |

| Chan-<br>nel | Register Name                               | Symbol* <sup>1</sup> | R/W     | Initial<br>Value | Address    | Access<br>Size |
|--------------|---|----------------------|---------|------------------|------------|----------------|
| В            | RSPI control register B                     | SPCRB                | R/W     | H'00             | H'FFFFB800 | 8, 16          |
|              | RSPI slave select polarity register B       | SSLPB                | R/W     | H'00             | H'FFFFB801 | 8              |
|              | RSPI pin control register B                 | SPPCRB               | R/W     | H'00             | H'FFFFB802 | 8, 16          |
|              | RSPI status register B                      | SPSRB                | R/(W)*2 | H'20             | H'FFFFB803 | 8              |
|              | RSPI data register B                        | SPDRB                | R/W     | H'0000           | H'FFFFB804 | 16             |
|              | RSPI sequence control register B            | SPSCRB               | R/W     | H'00             | H'FFFFB808 | 8, 16          |
|              | RSPI sequence status register B             | SPSSRB               | R       | H'00             | H'FFFFB809 | 8              |
|              | RSPI bit rate register B                    | SPBRB                | R/W     | H'FF             | H'FFFFB80A | 8              |
|              | RSPI clock delay register B                 | SPCKDB               | R/W     | H'00             | H'FFFFB80C | 8, 16          |
|              | RSPI slave select negation delay register B | SSLNDB               | R/W     | H'00             | H'FFFFB80D | 8              |
|              | RSPI next-access delay register B           | SPNDB                | R/W     | H'00             | H'FFFFB80E | 8              |
|              | RSPI command register B0                    | SPCMDB0              | R/W     | H'070D           | H'FFFFB810 | 16             |
|              | RSPI command register B1                    | SPCMDB1              | R/W     | H'070D           | H'FFFFB812 | 16             |
|              | RSPI command register B2                    | SPCMDB2              | R/W     | H'070D           | H'FFFFB814 | 16             |
|              | RSPI command register B3                    | SPCMDB3              | R/W     | H'070D           | H'FFFFB816 | 16             |
|              | RSPI command register B4                    | SPCMDB4              | R/W     | H'070D           | H'FFFFB818 | 16             |
|              | RSPI command register B5                    | SPCMDB5              | R/W     | H'070D           | H'FFFFB81A | 16             |
|              | RSPI command register B6                    | SPCMDB6              | R/W     | H'070D           | H'FFFFB81C | 16             |
|              | RSPI command register B7                    | SPCMDB7              | R/W     | H'070D           | H'FFFFB81E | 16             |

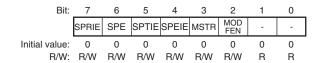
| Chan-<br>nel | Register Name                               | Symbol* <sup>1</sup> | R/W     | Initial<br>Value | Address    | Access<br>Size |
|--------------|---|----------------------|---------|------------------|------------|----------------|
| С            | RSPI control register C                     | SPCRC                | R/W     | H'00             | H'FFFFC000 | 8, 16          |
|              | RSPI slave select polarity register C       | SSLPC                | R/W     | H'00             | H'FFFFC001 | 8              |
|              | RSPI pin control register C                 | SPPCRC               | R/W     | H'00             | H'FFFFC002 | 8, 16          |
|              | RSPI status register C                      | SPSRC                | R/(W)*2 | H'20             | H'FFFFC003 | 8              |
|              | RSPI data register C                        | SPDRC                | R/W     | H'0000           | H'FFFFC004 | 16             |
|              | RSPI sequence control register C            | SPSCRC               | R/W     | H'00             | H'FFFFC008 | 8, 16          |
|              | RSPI sequence status register C             | SPSSRC               | R       | H'00             | H'FFFFC009 | 8              |
|              | RSPI bit rate register C                    | SPBRC                | R/W     | H'FF             | H'FFFFC00A | 8              |
|              | RSPI clock delay register C                 | SPCKDC               | R/W     | H'00             | H'FFFFC00C | 8, 16          |
|              | RSPI slave select negation delay register C | SSLNDC               | R/W     | H'00             | H'FFFFC00D | 8              |
|              | RSPI next-access delay register C           | SPNDC                | R/W     | H'00             | H'FFFFC00E | 8              |
|              | RSPI command register C0                    | SPCMDC0              | R/W     | H'070D           | H'FFFFC010 | 16             |
|              | RSPI command register C1                    | SPCMDC1              | R/W     | H'070D           | H'FFFFC012 | 16             |
|              | RSPI command register C2                    | SPCMDC2              | R/W     | H'070D           | H'FFFFC014 | 16             |
|              | RSPI command register C3                    | SPCMDC3              | R/W     | H'070D           | H'FFFFC016 | 16             |
|              | RSPI command register C4                    | SPCMDC4              | R/W     | H'070D           | H'FFFFC018 | 16             |
|              | RSPI command register C5                    | SPCMDC5              | R/W     | H'070D           | H'FFFFC01A | 16             |
|              | RSPI command register C6                    | SPCMDC6              | R/W     | H'070D           | H'FFFFC01C | 16             |
|              | RSPI command register C7                    | SPCMDC7              |         | H'070D           |            | 16             |

Notes: 1. Register names and symbols are used in the description for all channels, omitting the channel designation.

2. Only 0 can be written to clear the flag.

# 17.3.1 RSPI Control Register (SPCR)

SPCR sets the operating mode of the RSPI. SPCR can be read from or written to by the CPU. If the MSTR and MODFEN bits are changed while the RSPI function is enabled by setting the SPE bit to 1, subsequent operations cannot be guaranteed.



| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 7   | SPRIE    | 0                | R/W | RSPI Receive Interrupt Enable   |
|     |          |                  |     | If the RSPI has detected a receive buffer write after completion of a serial transfer and the SPRF bit in the RSPI status register (SPSR) is set to 1, this bit enables or disables the generation of an RSPI receive interrupt request.  |
|     |          |                  |     | <ol><li>Disables the generation of RSPI receive interrupt<br/>requests.</li></ol>   |
|     |          |                  |     | 1: Enables the generation of RSPI receive interrupt requests.   |
| 6   | SPE      | 0                | R/W | RSPI Function Enable  |
|     |          |                  |     | Setting this bit to 1 enables the RSPI function. When the MODF bit in the RSPI status register (SPSR) is 1, the SPE bit cannot be set to 1 (see section 17.4.7, Error Detection). Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function (see section 17.4.8, Initializing RSPI). |
|     |          |                  |     | 0: Disables the RSPI function   |
|     |          |                  |     | 1: Enables the RSPI function  |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 5   | SPTIE    | 0                | R/W | RSPI Transmit Interrupt Enable   |
|     |          |                  |     | Enables or disables the generation of RSPI transmit interrupt requests when the RSPI detects transmission buffer empty and sets the SPTEF bit in the RSPI status register (SPSR) to 1.   |
|     |          |                  |     | In the RSPI disabled (with the SPE bit 0) status, the SPTEF bit is 1. Therefore, note that setting the SPTIE bit to 1 when the RSPI is in the disabled status generates an RSPI transmit interrupt request.  0: Disables the generation of RSPI transmit interrupt requests.   |
|     |          |                  |     | Enables the generation of RSPI transmit interrupt requests.  |
| 4   | SPEIE    | 0                | R/W | RSPI Error Interrupt Enable  |
|     |          |                  |     | Enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault error and sets the MODF bit in the RSPI status register (SPSR) to 1, or when the RSPI detects and sets the OVRF bit in SPSR to 1 (see section 17.4.7, Error Detection). |
|     |          |                  |     | Disables the generation of RSPI error interrupt requests.  |
|     |          |                  |     | Enables the generation of RSPI error interrupt requests.   |
| 3   | MSTR     | 0                | R/W | RSPI Master/Slave Mode Select  |
|     |          |                  |     | Selects master/slave mode of RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCK, MOSI, MISO, and SSL1 to SSL7.  |
|     |          |                  |     | 0: Slave mode  |
|     |          |                  |     | 1: Master mode   |
| 2   | MODFEN   | 0                | R/W | Mode Fault Error Detection Enable  |
|     |          |                  |     | Enables or disables the detection of mode fault error (see section 17.4.7, Error Detection). In addition, the RSPI determines the input/output directions of the SSL0 pin based on combinations of the MODFEN and MSTR bits (see section 17.4.2, Controlling RSPI Pins).       |
|     |          |                  |     | 0: Disables the detection of mode fault error  |
|     |          |                  |     | 1: Enables the detection of mode fault error   |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 1, 0 | _        | All 0            | R   | Reserved   |
|      |          |                  |     | The write value should always be 0. Otherwise, operation cannot be guaranteed. |

### 17.3.2 RSPI Slave Select Polarity Register (SSLP)

SSLP sets the polarity of the SSL0 to SSL7 signals of the RSPI. SSLP can always be read from or written to by the CPU. If the contents of SSLP are changed by the CPU while the RSPI function is enabled by setting the SPE bit in the RSPI control register (SPCR) to 1, subsequent operations cannot be guaranteed.

Note that the SSL4 to SSL7 pins do not exist in channels B and C, therefore the polarity of the SSL pins in those channels cannot be switched by setting the SSL4P to SSL7P bits in the SSLPB and SSLPC registers.

| Bit:           | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | SSL7P | SSL6P | SSL5P | SSL4P | SSL3P | SSL2P | SSL1P | SSL0P |
| Initial value: | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R/W   |

| Bit | Bit Name | Initial<br>Value | R/W | Description                                     |
|-----|----------|------------------|-----|---|
| 7   | SSL7P    | 0                | R/W | SSL Signal Polarity Setting                     |
| 6   | SSL6P    | 0                | R/W | These bits set the polarity of the SSL signals. |
| 5   | SSL5P    | 0                | R/W | 0: SSLi signal 0-active                         |
| 4   | SSL4P    | 0                | R/W | 1: SSLi signal 1-active                         |
| 3   | SSL3P    | 0                | R/W |   |
| 2   | SSL2P    | 0                | R/W |   |
| 1   | SSL1P    | 0                | R/W | _   |
| 0   | SSL0P    | 0                | R/W |   |
|     |          |                  |     |   |

[Legend] i = 0 to 7

Page 754 of 1824

# 17.3.3 RSPI Pin Control Register (SPPCR)

SPPCR sets the modes of the RSPI pins. SPPCR can be read from or written to by the CPU. If the contents of this register are changed by the CPU while the RSPI function is enabled by setting the SPE bit in the RSPI control register (SPCR) to 1, operation cannot be guaranteed.

| Bit:           | 7 | 6 | 5     | 4     | 3 | 2    | 1 | 0    |
|----------------|---|---|-------|-------|---|------|---|------|
|                | - | - | MOIFE | MOIFV | - | SPOM | - | SPLP |
| Initial value: | 0 | 0 | 0     | 0     | 0 | 0    | 0 | 0    |
| R/W:           | R | R | R/W   | R/W   | R | R/W  | R | R/W  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 7, 6 | _        | All 0            | R   | Reserved  |
|      |          |                  |     | The write value should always be 0. Otherwise, operation cannot be guaranteed.  |
| 5    | MOIFE    | 0                | R/W | MOSI Idle Value Fixing Enable   |
|      |          |                  |     | Fixes the MOSI output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When MOIFE is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period. When MOIFE is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSI bit. |
|      |          |                  |     | 0: MOSI output value equals final data from previous transfer   |
|      |          |                  |     | MOSI output value equals the value set in the MOIFV bit   |
| 4    | MOIFV    | 0                | R/W | MOSI Idle Fixed Value   |
|      |          |                  |     | If the MOIFE bit is 1 in master mode, the RSPI, according to MOIFV bit settings, determines the MOSI signal value during the SSL negation period (including the SSL retention period during a burst transfer).  |
|      |          |                  |     | 0: MOSI Idle fixed value equals 0   |
|      |          |                  |     | 1: MOSI Idle fixed value equals 1   |
| 3    | _        | 0                | R   | Reserved  |
|      |          |                  |     | The write value should always be 0. Otherwise, operation cannot be guaranteed.  |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 2   | SPOM     | 0                | R/W | RSPI Output Pin Mode  |
|     |          |                  |     | Sets the RSPI output pins to CMOS output/open drain output.   |
|     |          |                  |     | 0: CMOS output  |
|     |          |                  |     | 1: Open-drain output  |
| 1   | _        | 0                | R   | Reserved  |
|     |          |                  |     | The write value should always be 0. Otherwise, operation cannot be guaranteed.  |
| 0   | SPLP     | 0                | R/W | RSPI Loopback   |
|     |          |                  |     | When the SPLP bit is set to 1, the RSPI shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects (reverses) the input path and the output path for the shift register. |
|     |          |                  |     | 0: Normal mode  |
| 1   |          |                  |     | 1: Loopback mode  |

# 17.3.4 RSPI Status Register (SPSR)

SPSR indicates the operating status of the RSPI. SPSR can be read by the CPU. Writing 1 to the SPRF, SPTEF, MODF, and OVRF bits cannot be performed by the CPU. These bits can be cleared to 0 after they are read as 1.

| Bit:           | 7      | 7 6 5 |        | 5 4 |   | 2      | 1 | 0      |
|----------------|--------|-------|--------|-----|---|--------|---|--------|
|                | SPRF   | -     | SPTEF  | -   | - | MODF   | - | OVRF   |
| Initial value: | 0      | 0     | 1      | 0   | 0 | 0      | 0 | 0      |
| R/W:           | R/(W)* | R     | R/(W)* | R   | R | R/(W)* | R | R/(W)* |

Note: \* Only 0 can be written to this bit after reading it as 1 to clear the flag.

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |  |  |  |
|-----|----------|------------------|--------|---|--|--|--|
|     |          |                  |        | <u> </u>  |  |  |  |
| 7   | SPRF     | 0                | R/(W)* | RSPI Receive Buffer Full Flag   |  |  |  |
|     |          |                  |        | Indicates the status of the receive buffer for the RSPI data register (SPDR). Upon completion of a serial transfer with the SPRF bit 0, the RSPI transfers the receive data from the shift register to SPDR, and sets this bit to 1. The SPRF bit is cleared to 0 under the following conditions: |  |  |  |
|     |          |                  |        | <ul> <li>The CPU reads SPSR when the SPRF bit is 1,<br/>and then the CPU writes a 0 to the SPRF bit.</li> </ul>   |  |  |  |
|     |          |                  |        | <ul> <li>The A-DMAC reads received data from the<br/>SPDR.</li> </ul>   |  |  |  |
|     |          |                  |        | <ul> <li>System power-on reset or standby</li> </ul>  |  |  |  |
|     |          |                  |        | • If a serial transfer ends while the SPRF bit is 1,  |  |  |  |
|     |          |                  |        | the RSPI does not transfer the received data  |  |  |  |
|     |          |                  |        | from the shift register to the SPDR. When the   |  |  |  |
|     |          |                  |        | OVRF bit in SPSR is 1, the SPRF bit cannot be   |  |  |  |
|     |          |                  |        | changed from 0 to 1 (see section 17.4.7, Error  |  |  |  |
|     |          |                  |        | Detection).   |  |  |  |
|     |          |                  |        | 0: No valid data in SPDR  |  |  |  |
|     |          |                  |        | 1: Valid data found in SPDR   |  |  |  |
| 6   | _        | 0                | R      | Reserved  |  |  |  |
|     |          |                  |        | The write value should always be 0. Otherwise, operation cannot be guaranteed.  |  |  |  |

| Bit  | Bit Name | Initial<br>Value | R/W    | Description   |
|------|----------|------------------|--------|---|
| 5    | SPTEF    | 1                | R/(W)* | RSPI Transmission Buffer Empty Flag   |
|      |          |                  |        | Indicates the status of the transmission buffer for the RSPI data register (SPDR). After the initialization of RSPI or after transmit data is transferred from the transmission buffer to the shift register, the RSPI sets the SPTEF bit to 1. The SPTEF bit is cleared to 0 under the following conditions. If the SPTEF bit is cleared and the shift register is empty, the data is copied from the transmission buffer to the shift register. |
|      |          |                  |        | • The CPU reads SPRF when the SPTEF bit is 1, and then the CPU writes 0 to the SPTEF bit.   |
|      |          |                  |        | The A-DMAC writes the transmit data to SPDR.  |
|      |          |                  |        | <ul> <li>The CPU and the A-DMAC can write to SPDR<br/>only when the SPTEF bit is 1. If the CPU or the<br/>A-DMAC writes to the transmission buffer of<br/>SPDR when the SPTEF bit is 0, the data in the<br/>transmission buffer is not updated.</li> </ul>  |
|      |          |                  |        | 0: Data found in the transmission buffer  |
|      |          |                  |        | 1: No data in the transmission buffer   |
| 4, 3 | _        | All 0            | R      | Reserved  |
|      |          |                  |        | The write value should always be 0. Otherwise, operation cannot be guaranteed.  |

| Bit     | Bit Name     | Initial<br>Value | R/W         | Description  |
|---------|--------------|------------------|-------------|--|
| 2       | MODF         | 0                | R/(W)*      | Mode Fault Error Flag  |
|         |              |                  |             | Indicates the occurrence of a mode fault error. When the input level of the SSL0 pin changes to the active level while the MSTR bit in the RSPI control register (SPCR) is 1 and the MODFEN bit is 1 with the RSPI being in multi-master mode, the RSPI detects a mode fault error and sets the MODF bit to 1. Similarly, if the MODFEN bit is set to 1 when the MSTR bit is 0 and the RSPI is in slave mode, and the SSL0 pin is negated before the RSPCK cycle necessary for data transfer ends, the RSPI detects a mode fault error. The active level of the SSL0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP). The MODF bit is cleared to 0 under the following conditions.  • The CPU reads SPSR when the MODF bit is 1, and then writes 0 to the MODF bit.  • Power-on reset or standby 0: No mode fault error occurs |
|         |              |                  |             | 1: A mode fault error occurs   |
| 1       | _            | 0                | R           | Reserved   |
|         |              |                  |             | The write value should always be 0. Otherwise, operation cannot be guaranteed.   |
| 0       | OVRF         | 0                | R/(W)*      | Overrun Error Flag   |
|         |              |                  |             | Indicates the occurrence of an overrun error. If a serial transfer ends while the SPRF bit is 1, the RSPI detects an overrun error, and sets the OVRF bit to 1. The OVRF bit is cleared to 0 under the following conditions.   |
|         |              |                  |             | <ul> <li>The CPU reads SPSR when the OVRF bit is 1,<br/>and then writes 0 to the OVRF bit.</li> </ul>  |
|         |              |                  |             | Power-on reset or standby  |
|         |              |                  |             | 0: No overrun error occurs   |
|         |              |                  |             | 1: An overrun error occurs   |
| Note: * | Only O can b | a writtan to     | thic hit of | ter reading it as 1 to clear the flag  |

Note: \* Only 0 can be written to this bit after reading it as 1 to clear the flag.

## 17.3.5 RSPI Data Register (SPDR)

SPDR is a buffer that stores RSPI transmit/receive data. The transmit and receive buffers are independent registers each other and are allocated to the same address.

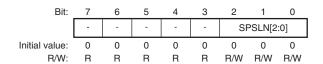
When the CPU or A-DMAC requests writing to SPDR and if the SPTEF bit in the RSPI status register (SPSR) is 1, the RSPI writes data to the transmission buffer of SPDR. If the SPTEF bit is 0, the RSPI does not update the SPDR transmission buffer.

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

| Bit:           | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|----------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
|                | SPD15 | SPD14 | SPD13 | SPD12 | SPD11 | SPD10 | SPD9 | SPD8 | SPD7 | SPD6 | SPD5 | SPD4 | SPD3 | SPD2 | SPD1 | SPD0 |
| Initial value: | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

# 17.3.6 RSPI Sequence Control Register (SPSCR)

SPSCR sets the sequence controlled method when the RSPI operates in master mode. SPSCR can be read from or written to by the CPU. If the contents of SPSCR are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function enabled, the subsequent operation cannot be guaranteed.



| Bit    | Bit Name   | Initial<br>Value | R/W | Descri   | ption              |   |  |  |
|--------|------------|------------------|-----|--|--------------------|---|--|--|
| 7 to 3 | _          | All 0            | R   | Reserv   | /ed                |   |  |  |
|        |            |                  |     |  |                    | hould always be 0. Otherwise,<br>be guaranteed.   |  |  |
| 2 to 0 | SPSLN[2:0] | 000              | R/W | RSPI S   | Sequence           | Length Setting  |  |  |
|        |            |                  |     | These bits set a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes RSPI command registers 0 to 7 (SPCMD0 to SPCMD7) to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN2 to SPSLN0 bits. When the RSPI is in slave mode, SPCMD0 is always referenced. |                    |   |  |  |
|        |            |                  |     | sequer   | •                  | among the setting in these bits,<br>, and referenced SPCMD register<br>ı below.   |  |  |
|        |            |                  |     | SPSLN<br>[2:0]   | Sequence<br>Length | Referenced SPCMD #  |  |  |
|        |            |                  |     | 000  | 1                  | $0 \rightarrow 0 \rightarrow \dots$   |  |  |
|        |            |                  |     | 001  | 2                  | $0 \rightarrow 1 \rightarrow 0 \rightarrow$   |  |  |
|        |            |                  |     | 010  | 3                  | $0 \rightarrow 1 \rightarrow 2 \rightarrow 0 \rightarrow$   |  |  |
|        |            |                  |     | 011  | 4                  | $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0 \rightarrow$   |  |  |
|        |            |                  |     | 100  | 5                  | $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0 \rightarrow$   |  |  |
|        |            |                  |     | 101  | 6                  | $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 0 \rightarrow$                                   |  |  |
|        |            |                  |     | 110  | 7                  | $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 0 \rightarrow$                     |  |  |
|        |            |                  |     | 111  | 8                  | $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 0 \rightarrow \dots$ |  |  |

# 17.3.7 RSPI Sequence Status Register (SPSSR)

SPSSR indicates the sequence control status when the RSPI operates in master mode. SPSSR can be read by the CPU. Any writing to SPSSR by the CPU is ignored.

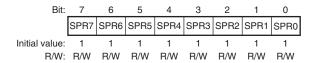
| Bit:           | 7 | 6  | 5     | 4    | 3 | 2 | 1      | 0  |
|----------------|---|----|-------|------|---|---|--------|----|
|                | - | SF | ECM[2 | 2:0] | - | S | PCP[2: | 0] |
| Initial value: | 0 | 0  | 0     | 0    | 0 | 0 | 0      | 0  |
| R/W:           | R | R  | R     | R    | R | R | R      | R  |

| Bit    | Bit Name   | Initial<br>Value | R/W | Description  |
|--------|------------|------------------|-----|--|
| 7      | _          | 0                | R   | Reserved   |
|        |            |                  |     | The write value should always be 0. Otherwise, operation cannot be guaranteed.   |
| 6 to 4 | SPECM[2:0] | 000              | R   | RSPI Error Command   |
|        |            |                  |     | These bits indicate RSPI command registers 0 to 7 (SPCMD0 to SPCMD7) that are pointed to by command pointers (SPCP2 to SPCP0 bits) when an error is detected during sequence control by the RSPI. The RSPI updates the bits SPECM2 to SPECM0 only when an error is detected. If both the OVRF and MODF bits in the RSPI status register (SPSR) are 0 and there is no error, the values of the bits SPECM2 to SPECM0 have no meaning. |
|        |            |                  |     | For the RSPI's error detection function, see section 17.4.7, Error Detection. For the RSPI's sequence control, see section 17.4.9, Master Mode Operation.  |
|        |            |                  |     | 000: SPCMD0  |
|        |            |                  |     | 001: SPCMD1  |
|        |            |                  |     | 010: SPCMD2  |
|        |            |                  |     | 011: SPCMD3  |
|        |            |                  |     | 100: SPCMD4  |
|        |            |                  |     | 101: SPCMD5  |
|        |            |                  |     | 110: SPCMD6  |
|        |            |                  |     | 111: SPCMD7  |
| 3      | _          | 0                | R   | Reserved   |
|        |            |                  |     | The write value should always be 0. Otherwise, operation cannot be guaranteed.   |

| Bit    | Bit Name  | Initial<br>Value | R/W | Description   |
|--------|-----------|------------------|-----|---|
| 2 to 0 | SPCP[2:0] | 000              | R   | RSPI Command Pointer  |
|        |           |                  |     | During RSPI sequence control, these bits indicate RSPI command registers 0 to 7 (SPCMD0 to SPCMD7), which are currently pointed to by the pointers. |
|        |           |                  |     | For the RSPI's sequence control, see section 17.4.9, Master Mode Operation.   |
|        |           |                  |     | 000: SPCMD0   |
|        |           |                  |     | 001: SPCMD1   |
|        |           |                  |     | 010: SPCMD2   |
|        |           |                  |     | 011: SPCMD3   |
|        |           |                  |     | 100: SPCMD4   |
|        |           |                  |     | 101: SPCMD5   |
|        |           |                  |     | 110: SPCMD6   |
|        |           |                  |     | 111: SPCMD7   |

## 17.3.8 RSPI Bit Rate Register (SPBR)

SPBR set the bit rate in master mode. SPBR can be read from or written to by the CPU. If the contents of SPBR are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.



The bit rate is determined by combinations of SPBR settings and the bit settings in the BRDV1 and BRDV0 bits in the RSPI command registers (SPCMD0 to SPCMD7). The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, ..., 255), and N denotes bit settings in the bits BRDV1 and BRDV0 (0, 1, 2, 3). The maximum bit rate is 10.0 Mbps in master mode. Normal operation cannot be guaranteed if the bit rate is set to exceed 10.0 Mbps.

Bit rate = 
$$\frac{f(P\phi)}{2 \times (n+1) \times 2^{N}}$$

Table 17.3 shows examples of the relationship between the SPBR register and BRDV1 and BRDV0 bit settings.

Table 17.3 Relationship between SPBR and BRDV1 and BRDV0 Settings

|          |               | Division | ion Bit Rate |           |                           |           |  |  |  |  |
|----------|---------------|----------|--------------|-----------|---------------------------|-----------|--|--|--|--|
| SPBR (n) | BRDV[1:0] (N) | ratio    | Pφ=16MHz     | Pφ=20MHz  | P <sub>\$\phi=32MHz</sub> | Pφ=40MHz  |  |  |  |  |
| 0        | 0             | 2        | 8.0 Mbps     | 10.0 Mbps |                           | _         |  |  |  |  |
| 1        | 0             | 4        | 4.0 Mbps     | 5.0 Mbps  | 8.0 Mbps                  | 10.0 Mbps |  |  |  |  |
| 2        | 0             | 6        | 2.67 Mbps    | 3.3 Mbps  | 5.33 Mbps                 | 6.67 Mbps |  |  |  |  |
| 3        | 0             | 8        | 2.0 Mbps     | 2.5 Mbps  | 4.0 Mbps                  | 5.0 Mbps  |  |  |  |  |
| 4        | 0             | 10       | 1.6 Mbps     | 2.0 Mbps  | 3.2 Mbps                  | 4.0 Mbps  |  |  |  |  |
| 5        | 0             | 12       | 1.33 Mbps    | 1.67 Mbps | 2.67 Mbps                 | 3.33 Mbps |  |  |  |  |
| 5        | 1             | 24       | 667 kbps     | 833 kbps  | 1.33 Mbps                 | 1.67 Mbps |  |  |  |  |
| 5        | 2             | 48       | 333 kbps     | 417 kbps  | 667 kbps                  | 8.33 kbps |  |  |  |  |
| 5        | 3             | 96       | 167 kbps     | 208 kbps  | 333 kbps                  | 417 kbps  |  |  |  |  |
| 255      | 3             | 4096     | 3.9 kbps     | 4.9 kbps  | 7.8 kbps                  | 9.8 kbps  |  |  |  |  |

Note: — setting prohibited

# 17.3.9 RSPI Clock Delay Register (SPCKD)

SPCKD sets a period from the beginning of SSL signal assertion to RSPCK oscillation (RSPCK delay) when the SCKDEN bit in the RSPI command register (SPCMD) is 1. SPCKD can be read from or written to by the CPU. If the contents of SPCKD are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2   | 1      | 0   |
|----------------|---|---|---|---|---|-----|--------|-----|
|                | - | - | - | - | - | SC  | CKDL[2 | :0] |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0   | 0      | 0   |
| R/W:           | R | R | R | R | R | R/W | R/W    | R/W |

| Bit    | Bit Name   | Initial<br>Value | R/W | Description  |
|--------|------------|------------------|-----|--|
| 7 to 3 | _          | All 0            | R   | Reserved   |
|        |            |                  |     | The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 2 to 0 | SCKDL[2:0] | 000              | R/W | RSPCK Delay Setting  |
|        |            |                  |     | These bits set an RSPCK delay value when the SCKDEN bit in SPCMD is 1.         |
|        |            |                  |     | 000: 1 RSPCK   |
|        |            |                  |     | 001: 2 RSPCK   |
|        |            |                  |     | 010: 3 RSPCK   |
|        |            |                  |     | 011: 4 RSPCK   |
|        |            |                  |     | 100: 5 RSPCK   |
|        |            |                  |     | 101: 6 RSPCK   |
|        |            |                  |     | 110: 7 RSPCK   |
|        |            |                  |     | 111: 8 RSPCK   |

# 17.3.10 SPI Slave Select Negation Delay Register (SSLND)

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSL signal during a serial transfer by the RSPI in master mode. SSLND can be read from or written to by the CPU. If the contents of SSLND are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2   | 1     | 0   |
|----------------|---|---|---|---|---|-----|-------|-----|
|                | - | - | - | - | - | SI  | NDL[2 | :0] |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0   | 0     | 0   |
| R/W:           | R | R | R | R | R | R/W | R/W   | R/W |

| Bit    | Bit Name   | Initial<br>Value | R/W | Description  |
|--------|------------|------------------|-----|--|
| 7 to 3 | _          | All 0            | R   | Reserved   |
|        |            |                  |     | The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 2 to 0 | SLNDL[2:0] | 000              | R/W | SSL Negation Delay Setting   |
|        |            |                  |     | These bits set an SSL negation delay value when the RSPI is in master mode.    |
|        |            |                  |     | 000: 1 RSPCK   |
|        |            |                  |     | 001: 2 RSPCK   |
|        |            |                  |     | 010: 3 RSPCK   |
|        |            |                  |     | 011: 4 RSPCK   |
|        |            |                  |     | 100: 5 RSPCK   |
|        |            |                  |     | 101: 6 RSPCK   |
|        |            |                  |     | 110: 7 RSPCK   |
|        |            |                  |     | 111: 8 RSPCK   |

# 17.3.11 RSPI Next-Access Delay Register (SPND)

SPND sets a non-active period (next-access delay) after termination of a serial transfer when the SPNDEN bit in the RSPI command register (SPCMD) is 1. SPND can be read from or written to by the CPU. If the contents of SPND are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2   | 1      | 0   |
|----------------|---|---|---|---|---|-----|--------|-----|
|                | - | - | - | - | - | SF  | PNDL[2 | :0] |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0   | 0      | 0   |
| R/W:           | R | R | R | R | R | R/W | R/W    | R/W |

| Bit    | Bit Name   | Initial<br>Value | R/W | Description  |
|--------|------------|------------------|-----|--|
| 7 to 3 | _          | All 0            | R   | Reserved   |
|        |            |                  |     | The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 2 to 0 | SPNDL[2:0] | 000              | R/W | RSPI Next-Access Delay Setting   |
|        |            |                  |     | These bits set a next-access delay when the SPNDEN bit in SPCMD is 1.          |
|        |            |                  |     | 000: 1 RSPCK + 2 Pφ  |
|        |            |                  |     | 001: 2 RSPCK + 2 Pφ  |
|        |            |                  |     | 010: 3 RSPCK + 2 Pφ  |
|        |            |                  |     | 011: 4 RSPCK + 2 Pφ  |
|        |            |                  |     | 100: 5 RSPCK + 2 Pφ  |
|        |            |                  |     | 101: 6 RSPCK + 2 Pφ  |
|        |            |                  |     | 110: 7 RSPCK + 2 Pφ  |
|        |            |                  |     | 111: 8 RSPCK + 2 Pφ  |

## 17.3.12 RSPI Command Register (SPCMD)

Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). SPCMD0 to SPCMD7 are used to set a transfer format for the RSPI in master mode. Some of the bits in SPCMD0 are used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references SPCMD0 to SPCMD7 according to the settings in bits SPSLN2 to SPSLN0 in the RSPI sequence control register (SPSCR), and executes the serial transfer that is set in the referenced SPCMD.

SPCMD can be read from or written to by the CPU. If the CPU rewrites the SPCMD that is referenced by the RSPI while the RSPI is performing serial transfer in master mode, or during SSL negation delay or next-access delay, correct operation cannot be guaranteed. SPCMD that is referenced by the RSPI in master mode can be checked by means of bits SPCP2 to SPCP0 in the RSPI sequence status register (SPSSR). When the RSPI function in slave mode is enabled, operation cannot be guaranteed if the value set in SPCMD0 is changed by the CPU.

| Bit:           | 15         | 14         | 13         | 12   | 11  | 10  | 9      | 8   | 7     | 6   | 5      | 4   | 3   | 2      | 1    | 0    |
|----------------|------------|------------|------------|------|-----|-----|--------|-----|-------|-----|--------|-----|-----|--------|------|------|
|                | SCK<br>DEN | SLN<br>DEN | SPN<br>DEN | LSBF |     | SPI | B[3:0] |     | SSLKP | S   | SLA[2: | 0]  | BRD | V[1:0] | CPOL | СРНА |
| Initial value: | 0          | 0          | 0          | 0    | 0   | 1   | 1      | 1   | 0     | 0   | 0      | 0   | 1   | 1      | 0    | 1    |
| R/W:           | R/W        | R/W        | R/W        | R/W  | R/W | R/W | R/W    | R/W | R/W   | R/W | R/W    | R/W | R/W | R/W    | R/W  | R/W  |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 15  | SCKDEN   | 0                | R/W | RSPCK Delay Setting Enable   |
|     |          |                  |     | Sets the period from the time the RSPI in master mode sets the SSL signal active until the RSPI oscillates RSPCK (RSPCK delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with RSPCK delay register (SPCKD) settings. |
|     |          |                  |     | To use the RSPI in slave mode, the SCKDEN bit should be set to 0.  |
|     |          |                  |     | 0: An RSPCK delay of 1 RSPCK   |
|     |          |                  |     | 1: An RSPCK delay equal to SPCKD settings.   |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 14  | SLNDEN   | 0                | R/W | SSL Negation Delay Setting Enable   |
|     |          |                  |     | Sets the period (SSL negation delay) from the time the master mode RSPI stops RSPCK oscillation until the RSPI sets the SSL signal inactive. If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with slave select negation delay register (SSLND) settings.  |
|     |          |                  |     | To use the RSPI in slave mode, the SLNDEN bit should be set to 0.   |
|     |          |                  |     | 0: An SSL negation delay of 1 RSPCK   |
|     |          |                  |     | 1: An SSL negation delay equal to SSLND settings.   |
| 13  | SPNDEN   | 0                | R/W | RSPI Next-Access Delay Enable   |
|     |          |                  |     | Sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSL signal inactive until the RSPI enables the SSL signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK + 2P $\varphi$ . If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with RSPI next-access delay register (SPND) settings. |
|     |          |                  |     | To use the RSPI in slave mode, the SPNDEN bit should be set to 0.   |
|     |          |                  |     | 0: A next-access delay of 1 RSPCK + 2 P $\phi$  |
|     |          |                  |     | 1: A next-access delay equal to SPND settings.  |
| 12  | LSBF     | 0                | R/W | RSPI LSB First  |
|     |          |                  |     | Sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.  |
|     |          |                  |     | 0: MSB first  |
|     |          |                  |     | 1: LSB first  |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 11 to 8 | SPB[3:0] | 0111             | R/W | SRPI Data Length Setting   |
|         |          |                  |     | These bits set a transfer data length for the RSPI in master mode or slave mode.   |
|         |          |                  |     | 0000 to 0111: 8 bits   |
|         |          |                  |     | 1000: 9 bits   |
|         |          |                  |     | 1001: 10 bits  |
|         |          |                  |     | 1010: 11 bits  |
|         |          |                  |     | 1011: 12 bits  |
|         |          |                  |     | 1100: 13 bits  |
|         |          |                  |     | 1101: 14 bits  |
|         |          |                  |     | 1110: 15 bits  |
|         |          |                  |     | 1111: 16 bits  |
| 7       | SSLKP    | 0                | R/W | SSL Signal Level Keeping   |
|         |          |                  |     | When the RSPI in master mode performs a serial transfer, this bit specifies whether the SSL signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command. |
|         |          |                  |     | To use the RSPI in slave mode, the SSLKP bit should be set to 0.   |
|         |          |                  |     | <ol><li>Negates all SSL signals upon completion of<br/>transfer.</li></ol>   |
|         |          |                  |     | Keeps the SSL signal level from the end of the transfer until the beginning of the next access.  |

| Bit    | Bit Name  | Initial<br>Value | R/W | Description  |
|--------|-----------|------------------|-----|--|
| 6 to 4 | SSLA[2:0] | 000              | R/W | SSL Signal Assertion Setting   |
|        |           |                  |     | These bits control the SSL signal assertion when the RSPI performs serial transfers in master mode. Setting the SSLA[2:0] controls the assertion for the signals SSL7 to SSL0. When an SSL signal is asserted, its polarity is determined by the set value in the corresponding SSLP (RSPI slave select polarity register). When SSLA[2:0] are set to B'000 in multimaster mode, serial transfers are performed with all the SLL signals in the neagted state (as SSL0 acts as input). |
|        |           |                  |     | Also when setting the values B'100 to B'111 to the bits SSLA[2:0] of the registers SPCMDB0 to SPCMDB7 and SPCMDC0 to SPCMDC7, serial transfers are performed with the SSL signals in the negated state (as channels B and C lack SSL4 to SSL7).  |
|        |           |                  |     | When using the RSPI in slave mode, set B'000 to SSLA[2:0].   |
|        |           |                  |     | 000: SSL0  |
|        |           |                  |     | 001: SSL1  |
|        |           |                  |     | 010: SSL2  |
|        |           |                  |     | 011: SSL3  |
|        |           |                  |     | 100: SSL4  |
|        |           |                  |     | 101: SSL5  |
|        |           |                  |     | 110: SSL6  |
|        |           |                  |     | 111: SSL7  |

| Bit  | Bit Name  | Initial<br>Value | R/W | Description   |
|------|-----------|------------------|-----|---|
| 3, 2 | BRDV[1:0] | 11               | R/W | Bit Rate Division Setting   |
|      |           |                  |     | These bits are used to determine the bit rate. A bit rate is determined by combinations of bits BRDV1 and BRDV 0 and the settings in the RSPI bit rate register (SPBR) (see section 17.3.8, RSPI Bit Rate Register (SPBR)). The settings in SPBR determine the base bit rate. The settings in bits BRDV1 and BRDV0 are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In the bits SPCMD0 to SPCMD7 different BRDV1 and BRDV0 settings can be specified. This permits the execution of serial transfers at a different bit rate for each command. |
|      |           |                  |     | 00: Select the base bit rate  |
|      |           |                  |     | 01: Select the base bit rate divided by 2   |
|      |           |                  |     | 10: Select the base bit rate divided by 4   |
|      |           |                  |     | 11: Select the base bit rate divided by 8   |
| 1    | CPOL      | 0                | R/W | RSPCK Polarity Setting  |
|      |           |                  |     | Sets an RSPCK polarity of the RSPI in master or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules.   |
|      |           |                  |     | 0: RSPCK = 0 when idle  |
|      |           |                  |     | 1: RSPCK = 1 when idle  |
| 0    | СРНА      | 1                | R/W | RSPCK Phase Setting   |
|      |           |                  |     | Sets an RSPCK phase of the RSPI in master or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules.   |
|      |           |                  |     | Data sampling on odd edge, data variation on even edge  |
|      |           |                  |     | Data variation on odd edge, data sampling on even edge  |

# 17.4 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

# 17.4.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave, single-master, and multi-master modes. A particular mode of the RSPI can be selected by using the MSTR and MODFEN bits in the RSPI control register (SPCR). Table 17.4 gives the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 17.4 Relationship between RSPI Modes and SPCR and Description of Each Mode

| Item                         |  | Slave Mode      | Single-Master Mode          | Multi-Master Mode           |
|------------------------------|--|-----------------|-----------------------------|-----------------------------|
| MSTR bit setting             |  | 0               | 1                           | 1                           |
| MODFEN b                     | oit setting                                    | 0, 1            | 0                           | 1                           |
| RSPCK sig                    | nal  | Input           | Output                      | Output/Hi-Z                 |
| MOSI signa                   | al   | Input           | Output                      | Output/Hi-Z                 |
| MISO signa                   | al   | Output/Hi-Z     | Input                       | Input                       |
| SSL0 signa                   | ıl   | Input           | Output                      | Input                       |
| SSL1 to SS                   | L7 signals                                     | Hi-Z            | Output                      | Output/Hi-Z                 |
| Output pin                   | mode   | CMOS/open-drain | CMOS/open-drain             | CMOS/open-drain             |
| SSL polarity<br>modification | •  | Supported       | Supported                   | Supported                   |
| Transfer rate                | One-time<br>multiplied<br>peripheral<br>clock  | Up to 2.5 MHz   | Up to 10.0 MHz              | Up to 10.0 MHz              |
|                              | Two-times<br>multiplied<br>peripheral<br>clock | Up to 2.5 MHz   | Up to 10.0 MHz              | Up to 10.0 MHz              |
| Clock source                 |  | RSPCK input     | On-chip baud rate generator | On-chip baud rate generator |
| Clock polarity               |  | Two             | Two                         | Two                         |
| Clock phase                  |  | Two             | Two                         | Two                         |
| First transfe                | er bit   | MSB/LSB         | MSB/LSB                     | MSB/LSB                     |

| Item                                | Slave Mode                            | Single-Master Mode                       | Multi-Master Mode                        |
|-------------------------------------|---------------------------------------|--|--|
| Transfer data length                | 8 to 16 bits                          | 8 to 16 bits                             | 8 to 16 bits                             |
| Burst transfer                      | Possible (CPHA = 1)                   | Possible (CPHA = 0, 1)                   | Possible (CPHA = 0, 1)                   |
| RSPCK delay control                 | Not supported                         | Supported                                | Supported                                |
| SSL negation delay control          | Not supported                         | Supported                                | Supported                                |
| Next-access delay control           | Not supported                         | Supported                                | Supported                                |
| Transfer starting method            | SSL input active or RSPCK oscillation | Started by the A-<br>DMAC: write to SPDR | Started by the A-<br>DMAC: write to SPDR |
|                                     |                                       | Started by the CPU: clear SPTEF          | Started by the CPU: clear SPTEF          |
| Sequence control                    | Not supported                         | Supported                                | Supported                                |
| Transmission buffer empty detection | Supported                             | Supported                                | Supported                                |
| Receive buffer full detection       | Supported                             | Supported                                | Supported                                |
| Overrun error detection             | Supported                             | Supported                                | Supported                                |
| Mode fault error detection          | Supported<br>(MODFEN = 1)             | Not supported                            | Supported                                |

Pin State\*1

Input Hi-Z

Input

Open-drain output/Hi-Z

#### **Controlling RSPI Pins** 17.4.2

According to the MSTR and MODFEN bits in the RSPI control register (SPCR) and the SPOM bit in the RSPI pin control register (SPPCR), the RSPI can automatically switch pin directions and output modes. Table 17.5 shows the relationship between pin states and bit settings.

Table 17.5 Relationship between Pin States and Bit Settings

| Mode                   | Pin                           | SPOM = 0             | SPOM = 1               |
|------------------------|-------------------------------|----------------------|------------------------|
| Single-master mode     | RSPCK                         | CMOS output          | Open-drain output      |
| (MSTR = 1, MODFEN = 0) | SSL0 to SSL7 CMOS output Oper |                      | Open-drain output      |
|                        | MOSI                          | MOSI CMOS output Ope |                        |
|                        | MISO                          | Input                | Input                  |
| Multi-master mode      | RSPCK*2                       | CMOS output/Hi-Z     | Open-drain output/Hi-Z |
| (MSTR = 1, MODFEN = 1) | SSL0                          | Input                | Input                  |
|                        | SSL1 to SSL7                  | CMOS output/Hi-Z     | Open-drain output/Hi-Z |
|                        | MOSI*2                        | CMOS output/Hi-Z     | Open-drain output/Hi-Z |
|                        | MISO                          | Input                | Input                  |
| Slave mode (MSTR = 0)  | RSPCK                         | Input                | Input                  |

Notes: 1. RSPI settings are not indicated in the multiplex pins for which the RSPI function is not selected.

2. When SSL0 is at the active level, the pin state is Hi-Z.

SSL1 to SSL7

SSL<sub>0</sub>

MOSI

MISO\*3

3. When SSL0 is at the non-active level or the SPE bit in SPCR is 0, the pin state is Hi-Z.

Input

Hi-Z

Input

CMOS output/Hi-Z

The RSPI in master mode determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as shown in table 17.6.

Table 17.6 MOSI Signal Value Determination during SSL Negation Period

| MOIFE | MOIFV | MOSI Signal Value during SSL Negation Period |
|-------|-------|--|
| 0     | 0, 1  | Final data from previous transfer            |
| 1     | 0     | Always 0                                     |
| 1     | 1     | Always 1                                     |

## 17.4.3 RSPI System Configuration Example

### (1) Single Master/Single Slave (with this LSI Acting as Master)

Figure 17.2 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a master. In the single-master/single-slave configuration, the SSL0 to SSL7 output of this LSI (master) is not used. The SSL input of the RSPI slave is fixed to the low level, and the RSPI slave is always maintained in a select state. In the transfer format corresponding to the case where the CPHA bit in the RSPI control register (SPCR) is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSL output of this LSI should be connected to the SSL input of the slave device.

This LSI (master) always drives the RSPCK and MOSI. The RSPI slave always drives the MISO.

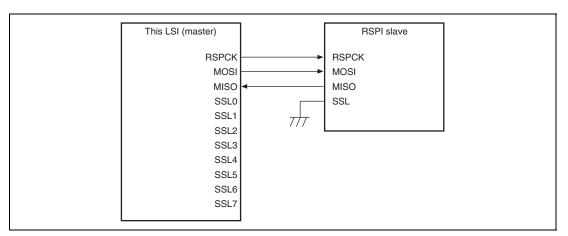


Figure 17.2 Single-Master/Single-Slave Configuration Example (This LSI = Master)

#### (2)Single Master/Single Slave (with this LSI Acting as Slave)

Figure 17.3 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, the SSL0 pin is used as SSL input. The RSPI master always drives the RSPCK and MOSI. This LSI (slave) always drives the MISO.

In the single-slave configuration in which the CPHA bit in the RSPI command register (SPCMD) is set to 1, the SSL0 input of this LSI (slave) is fixed to the low level, this LSI (slave) is always maintained in a selected state, and in this manner it is possible to execute serial transfer (figure 17.4).

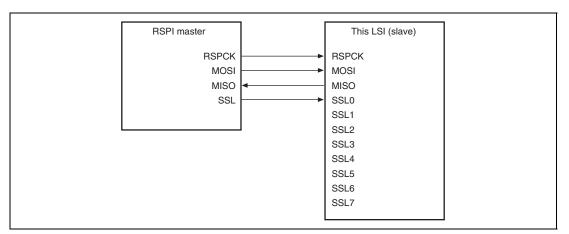


Figure 17.3 Single-Master/Single-Slave Configuration Example (This LSI = Slave)

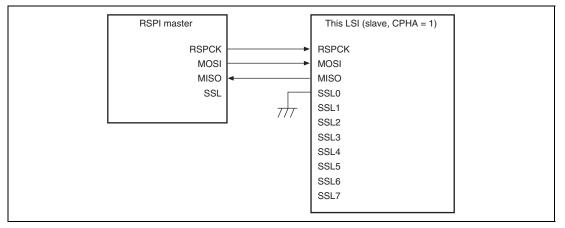


Figure 17.4 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1)

## (3) Single Master/Multi-Slave (with this LSI Acting as Master)

Figure 17.5 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of figure 17.5, the RSPI system is comprised of this LSI (master) and four slaves (RSPI slave 0 to RSPI slave 3).

The RSPCK and MOSI outputs of this LSI (master) are connected to the RSPCK and MOSI inputs of RSPI slave 0 to RSPI slave 3. The MISO outputs of RSPI slave 0 to RSPI slave 3 are all connected to the MISO input of this LSI (master). SSL0 to SSL3 outputs of this LSI (master) are connected to the SSL inputs of RSPI slave 0 to RSPI slave 3, respectively. In this configuration example, because there are four RSPI slaves, the SSL4 to SSL7 outputs of this LSI (master) are not used.

This LSI (master) always drives RSPCK, MOSI, and SSL0 to SSL3. Of the RSPI slave 0 to RSPI slave 3, the slave that receives low level input into the SSL input drives MISO.

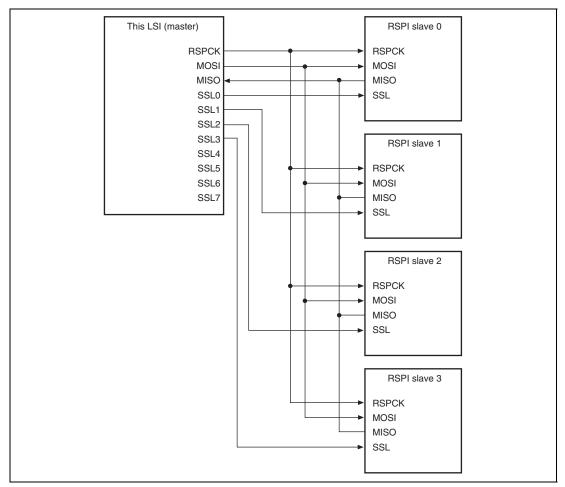


Figure 17.5 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

### (4) Single Master/Multi-Slave (with this LSI Acting as Slave)

Figure 17.6 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a slave. In the example of figure 17.6, the RSPI system is comprised of an RSPI master and the two LSIs (slave X and slave Y).

The RSPCK and MOSI outputs of the RSPI master are connected to the RSPCK and MOSI inputs of the LSIs (slave X and slave Y). The MISO outputs of the LSIs (slave X and slave Y) are all connected to the MISO input of the RSPI master. SSLX and SSLY outputs of the RSPI master are connected to the SSL0 inputs of the LSIs (slave X and slave Y), respectively.

The RSPI master always drives RSPCK, MOSI, SSLX, and SSLY. Of the LSIs (slave X and slave Y), the slave that receives low level input into the SSL0 input drives MISO.

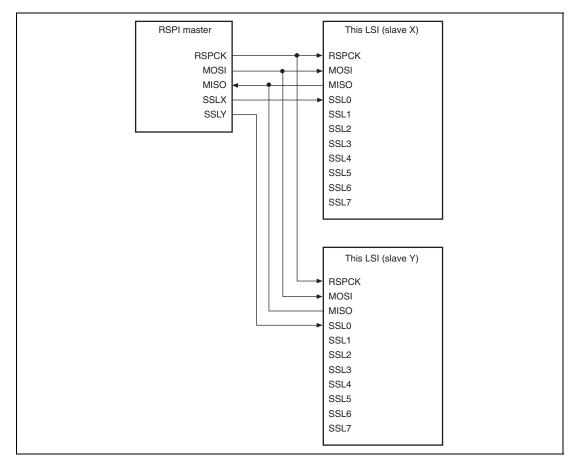


Figure 17.6 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

#### Multi-Master/Multi-Slave (with this LSI Acting as Master) (5)

Figure 17.7 shows a multi-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of figure 17.7, the RSPI system is comprised of the two LSIs (master X, master Y) and two RSPI slaves (RSPI slave 1, RSPI slave 2).

The RSPCK and MOSI outputs of the LSIs (master X, master Y) are connected to the RSPCK and MOSI inputs of RSPI slaves 1 and 2. The MISO outputs of RSPI slaves 1 and 2 are connected to the MISO inputs of the LSIs (master X, master Y). Any generic port Y output from this LSI (master X) is connected to the SSL0 input of this LSI (master Y). Any generic port X output of this LSI (master Y) is connected to the SSL0 input of this LSI (master X). The SSL1 and SSL2 outputs of the LSIs (master X, master Y) are connected to the SSL inputs of the RSPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSL0 input, and SSL1 and SSL2 outputs for slave connections, the outputs SSL3 to SSL7 of this LSI are not required.

This LSI drives RSPCK, MOSI, SSL1, and SSL2 when the SSL0 input level is 1. When the SSL0 input level is 0, this LSI detects a mode fault error, sets RSPCK, MOSI, SSL1, and SSL2 to Hi-Z, and releases the RSPI bus right to the other master. Of the RSPI slaves 1 and 2, the slave that receives a level-0 input into the SSL input drives MISO.

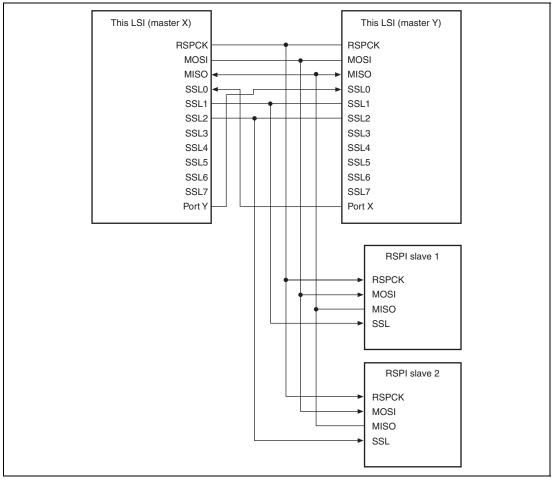


Figure 17.7 Multi-Master/Multi-Slave Configuration Example (This LSI = Master)

### 17.4.4 Transfer Format

### (1) CPHA = 0

Figure 17.8 shows an example transfer format for the serial transfer of 8-bit data when the CPHA bit in the RSPI command register (SPCMD) is 0. In Figure 17.8, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI settings. For details, see section 17.4.2, Controlling RSPI Pins.

When the CPHA bit is 0, the driving of valid data to the MOSI and MISO signals commences at an SSL signal assertion timing. The first RSPCK signal change timing that occurs after the SSL signal assertion becomes the first transfer data fetching timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSI and MISO signals is always 1/2 RSPCK cycle after the transfer data fetch timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1 denotes a period from an SSL signal assertion to RSPCK oscillation (RSPCK delay). t2 denotes a period from the cessation of RSPCK oscillation to an SSL signal negation (SSL negation delay). t3 denotes a period in which SSL signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 17.4.9, Master Mode Operation.

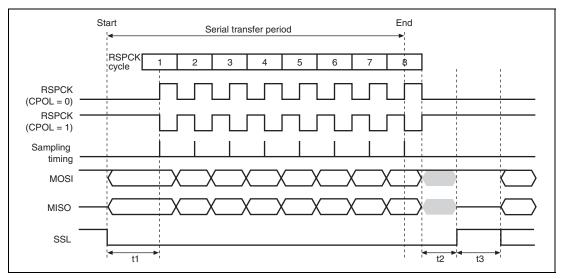


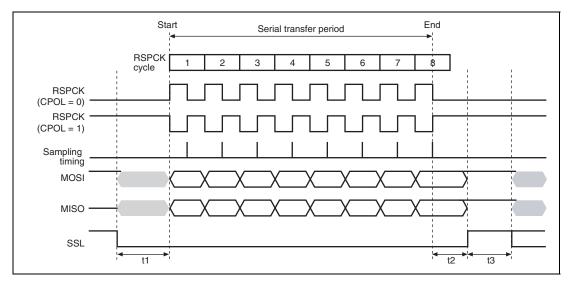
Figure 17.8 RSPI Transfer Format (CPHA = 0)

## $(2) \quad CPHA = 1$

Figure 17.9 shows an example transfer format for the serial transfer of 8-bit data when the CPHA bit in the RSPI command register (SPCMD) is 1. In figure 17.9, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI modes (master or slave). For details, see section 17.4.2, Controlling RSPI Pins.

When the CPHA bit is 1, the driving of invalid data to the MOSI and MISO signals commences at an SSL signal assertion timing. The driving of valid data to the MOSI and MISO signals commences at the first RSPCK signal change timing that occurs after the SSL signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycle after the data update timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 17.4.9, Master Mode Operation.



**Figure 17.9 RSPI Transfer Format (CPHA = 1)** 

### 17.4.5 Data Format

The RSPI's data format depends on the settings in the RSPI command register (SPCMD). Irrespective of MSB/LSB first, the RSPI treats the range from the LSB of the RSPI data register (SPDR) to the assigned data length as transfer data.

### (1) MSB First Transfer (16-Bit Data)

Figure 17.10 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 16-bit data length MSB-first data transfer.

The CPU or the A-DMAC writes T15 to T00 to the transmission buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI copies the data in the transmission buffer of SPDR to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 15) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 16 bits has passed, data R15 to R00 is stored in the shift register. In this state, the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the A-DMAC writes to the transmission buffer of SPDR, received data R15 to R00 is shifted out from the shift register.

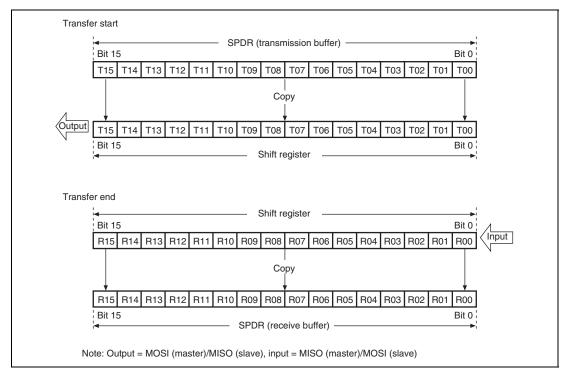


Figure 17.10 MSB First Transfer (16-Bit Data)

### (2) MSB First Transfer (10-Bit Data)

Figure 17.11 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 10-bit data length MSB-first data transfer.

The CPU or the A-DMAC writes T15 to T00 to the transmission buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI copies the data in the transmission buffer of SPDR to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from bit 9 of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 10 bits has passed, received data R09 to R00 is stored in bits 9 to 0 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 15 to 10 in the shift register. In this state, the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the A-DMAC writes to the transmission buffer of SPDR, received data R09 to R00 is shifted out from the shift register.

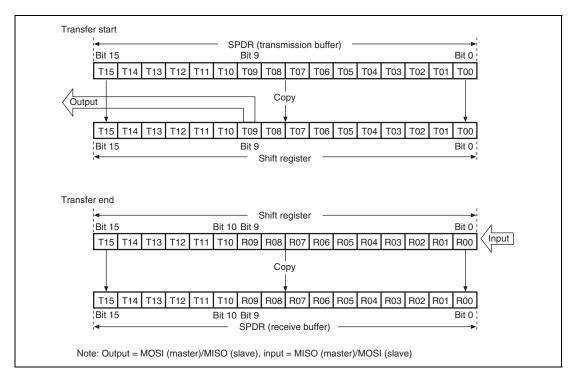


Figure 17.11 MSB First Transfer (10-Bit Data)

### (3) LSB First Transfer (16-Bit Data)

Figure 17.12 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 16-bit data length LSB-first data transfer.

The CPU or the A-DMAC writes T15 to T00 to the transmission buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI reverses the order of the bits of the data in the transmission buffer of SPDR, copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 15) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 16 bits has passed, data R00 to R15 is stored in the shift register. In this state, the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the A-DMAC writes to the transmission buffer of SPDR, received data R00 to R15 is shifted out from the shift register.

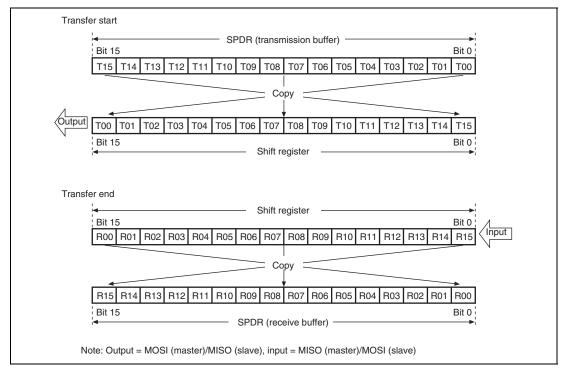


Figure 17.12 LSB First Transfer (16-Bit Data)

#### **(4)** LSB First Transfer (10-Bit Data)

Figure 17.13 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 10-bit data length LSB-first data transfer.

The CPU or the A-DMAC writes T15 to T00 to the transmission buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI reverses the order of the bits of the data in the transmission buffer of SPDR, copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 15) of the shift register, and shifts in the data from bit 6 of the shift register. When the RSPCK cycle required for the serial transfer of 10 bits has passed, received data R00 to R09 is stored in bits 15 to 6 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 5 to 0 of the shift register. In this state, the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the A-DMAC writes to the transmission buffer of SPDR, received data R00 to R09 is shifted out from the shift register.

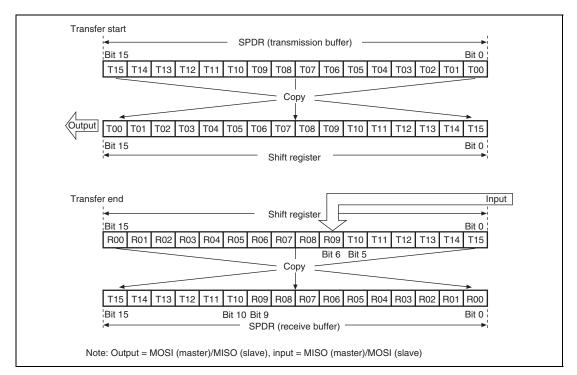


Figure 17.13 LSB First Transfer (10-Bit Data)

### 17.4.6 Transmission Buffer Empty/Receive Buffer Full Flags

Figure 17.14 shows an example of operation of the RSPI transmission buffer empty flag (SPTEF) and the RSPI receive buffer full flag in the RSPI status register (SPSR). The SPDR access depicted in figure 17.14 indicates the condition of access from the A-DMAC to the RSPI data register (SPDR), where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of figure 17.14, the RSPI performs an 8-bit serial transfer in which the CPHA bit in the RSPI command register (SPCMD) is 1, and CPOL is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

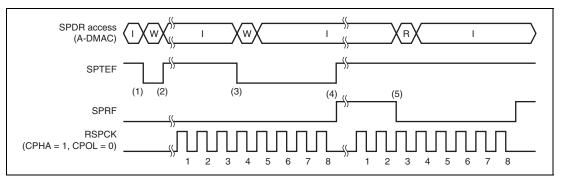


Figure 17.14 SPTEF and SPRF Bit Operation Example

The operation of the flags at timings shown in steps (1) to (5) in the figure is described below.

- 1. When the A-DMAC writes transmit data to SPDR when the transmission buffer of SPDR is empty, the RSPI sets the SPTEF bit to 0, and writes data to the transmission buffer, with no change in the SPRF flag.
- 2. If the shift register is empty, the RSPI sets the SPTEF bit to 1, and copies the data in the transmission buffer to the shift register, with no change in the SPRF flag. How a serial transfer is started depends on the mode of the RSPI. For details, see section 17.4.9, Master Mode Operation, and section 17.4.10, Slave Mode Operation.
- 3. When the A-DMAC writes transmit data to SPDR with the transmission buffer of SPDR being empty, the RSPI sets the SPTEF bit to 1, and writes data to the transmission buffer, while the SPRF flag remains unchanged. Because the data being transferred serially is stored in the shift register, the RSPI does not copy the data in the transmission buffer to the shift register.

- 4. When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI sets the SPRF bit to 1, and copies the receive data in the shift register to the receive buffer. Because the shift register becomes empty upon completion of serial transfer, if the transmission buffer was full before the serial transfer ended, the RSPI sets the SPTEF bit to 1, and copies the data in the transmission buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer the RSPI determines that the shift register is empty, and as a result data transfer from the transmission buffer to the shift register is enabled.
- 5. When the A-DMAC reads SPDR with the receive buffer being full, the RSPI sets the SPRF bit to 0, and sends the data in the receive buffer to the bus inside the chip.

If the CPU or the A-DMAC writes to SPDR when the SPTEF bit is 0, the RSPI does not update the data in the transmission buffer. When writing to SPDR, make sure that the SPTEF bit is 1. That the SPTEF bit is 1 can be checked by reading SPSR or by using an RSPI transmit interrupt. To use an RSPI transmit interrupt, set the SPTIE bit in SPCR to 1.

If the RSPI is disabled (the SPE bit in SPCR being 0), the SPTEF bit is initialized to 1. For this reason, setting the SPTIE bit to 1 when the RSPI is disabled generates an RSPI transmit interrupt.

When serial transfer ends with the SPRF bit being 1, the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (see section 17.4.7, Error Detection). To prevent a receive data overrun error, set the SPRF bit to 0 before the serial transfer ends. That the SPRF bit is 1 can be checked by either reading SPSR or by using an RSPI receive interrupt. To use an RSPI receive interrupt, set the SPRIE bit in SPCR to 1.

### 17.4.7 Error Detection

In the normal RSPI serial transfer, the data written from the RSPI data register (SPDR) to the transmission buffer by either the CPU or the A-DMAC is serially transmitted, and either the CPU or the A-DMAC can read the serially received data from the receive buffer of SPDR. If access is made to SPDR by either the CPU or the A-DMAC, depending on the status of the transmission buffer/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error or a mode fault error. Table 17.7 shows the relationship between non-normal transfer operations and the RSPI's error detection function.

Relationship between Non-Normal Transfer Operations and RSPI Error **Table 17.7 Detection Function** 

|   | Occurrence Condition   | RSPI Operation   | Error Detection  |
|---|--|--|------------------|
| A | Either the CPU or the A-DMAC writes to SPDR when the transmission buffer is full.                      | Retains the contents of the transmission buffer. Missing write data.     | None             |
| В | Serial transfer is started in slave mode when transmit data is still not loaded on the shift register. | Data received in previous serial transfer is serially transmitted.       | None             |
| С | Either the CPU or the A-DMAC reads from SPDR when the receive buffer is empty.                         | Previously received serial data is output to the CPU or the A-DMAC.      | None             |
| D | Serial transfer terminates when the receive buffer is full.  | Retains the contents of the receive buffer. Missing serial receive data. | Overrun error    |
| E | The SSL0 input signal is asserted when the serial transfer is idle in multi-master mode.               | RSPI disabled.   | Mode fault error |
| F | The SSL0 input signal is asserted  | Serial transfer suspended.   | Mode fault error |
|   | during serial transfer in multi-master mode.   | Missing send/receive data.   |                  |
|   | mode.  | Driving of the RSPCK, MOSI, and SSL1 to SSL7 output signals stopped.     |                  |
|   |  | RSPI disabled.   |                  |
| G | The SSL0 input signal is negated   | Serial transfer suspended.   | Mode fault error |
|   | during serial transfer in slave mode.  | Missing send/receive data.   |                  |
|   |  | Driving of the MISO output signal stopped.                               |                  |
|   |  | RSPI disabled.   |                  |

On operation A shown in table 17.7, the RSPI does not detect an error. To prevent data omission during the writing to SPDR by the CPU or the A-DMAC, write operations to SPDR should be executed when the SPTEF bit in the RSPI status register (SPSR) is 1.

Likewise, the RSPI does not detect an error on operation B. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in B as an error. Notice that the received data from the previous serial transfer is retained in the receive buffer of SPDR, and thus it can be

correctly read by the CPU or the A-DMAC (if SPDR is not read before the end of the serial transfer, an overrun error may result).

Similarly, the RSPI does not detect an error on operation C. To prevent the CPU or the A-DMAC from reading extraneous data, SPDR read operation should be executed when the SPRF bit in SPSR is 1.

An overrun error shown in D is described in section 17.4.7 (1), Overrun Error. A mode fault error shown in E to G is described in section 17.4.7 (2), Mode Fault Error. On operations of the SPTEF and SPRF bits in SPSR, see section 17.4.6, Transmission Buffer Empty/Receive Buffer Full Flags.

### (1) Overrun Error

If serial transfer ends when the receive buffer of the RSPI data register (SPDR) is full, the RSPI detects an overrun error, and sets the OVRF bit in SPSR to 1. When the OVRF bit is 1, the RSPI does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To reset the OVRF bit in SPSR to 0, either execute power-on reset/standby, or write a 0 to the OVRF bit after the CPU has read SPSR with the OVRF bit set to 1.

Figure 17.15 shows an example of operation of the SPRF and OVRF bits in SPSR. The SPSR access depicted in figure 17.15 indicates the condition of access from the CPU to SPSR, and from the A-DMAC to SPDR, respectively, where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of figure 17.15, the RSPI performs an 8-bit serial transfer in which the CPHA bit in the RSPI command register (SPCMD) is 1, and CPOL is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

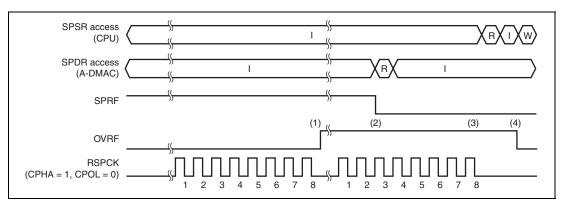


Figure 17.15 SPRF and OVRF Bit Operation Example

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

- 1. If a serial transfer terminates with the SPRF bit being 1 (receive buffer full), the RSPI detects an overrun error, and sets the OVRF bit to 1. The RSPI does not copy the data in the shift register to the receive buffer. In master mode, the RSPI copies the value of the pointer to the RSPI command register (SPCMD) to bits SPECM2 to SPECM0 in the RSPI sequence status register (SPSSR).
- 2. When the A-DMAC reads SPDR, the RSPI sets the SPRF bit to 0, and outputs the data in the receive buffer to an internal bus. The receive buffer becoming empty does not clear the OVRF hit.
- 3. If the serial transfer terminates with the OVRF bit being 1 (an overrun error), the RSPI keeps the SPRF bit at 0 and does not update it. Likewise, the RSPI does not copy the data in the shift register to the receive buffer. When in master mode, the RSPI does not update bits SPECM2 to SPECM0 of SPSSR. If, in an overrun error state, the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer is enabled from the transmission buffer to the shift register.
- 4. If the CPU writes a 0 to the OVRF bit after reading SPSR when the OVRF bit is 1, the RSPI clears the OVRF bit.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When using an RSPI error interrupt, set the SPEIE bit in the RSPI control register (SPCR) to 1. When executing a serial transfer without using an RSPI error interrupt, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is run in master mode, the pointer value to SPCMD can be checked by reading bits SPECM2 to SPECM0 of SPSSR.

If an overrun error occurs and the OVRF bit is set to 1, normal reception operations cannot be performed until such time as the OVRF bit is cleared. The OVRF bit is cleared to 0 under the following conditions:

- After reading SPSR in a condition in which the OVRF bit is set to 1, the CPU writes a 0 to the OVRF bit.
- Power-on reset/standby

### (2) Mode Fault Error

The RSPI operates in multi-master mode when the MSTR bit in the RSPI control register (SPCR) is 1 and the MODFEN bit is also 1. If the active level is input with respect to the SSL0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the MODF bit in the RSPI status register (SPSR) to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to the RSPI command register (SPCMD) to bits SPECM2 to SPECM0 in the RSPI sequence status register (SPSSR). The active level of the SSL0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP).

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit in the RSPI in slave mode is 1, and if the SSL0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops the driving of output signals and clears the SPE bit in the SPCR register. When the SPE bit is cleared, the RSPI function is disabled (see section 17.4.8, Initializing RSPI). In multi-master configuration, it is possible to release the master right by using a mode fault error to stop the driving of output signals and the RSPI function.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When using an RSPI error interrupt, set the SPEIE bit in the RSPI control register (SPCR) to 1. To detect a mode fault error without using an RSPI error interrupt, it is necessary to poll SPSR. When using the RSPI in master mode, one can read bits SPECM2 to SPECM0 of SPSSR to verify the value of the pointer to SPCMD when an error occurs.

When the MODF bit is 1, the RSPI ignores the writing of the value 1 to the SPE bit by the CPU. To enable the RSPI function after the detection of a mode fault error, the MODF bit must be set to 0. The MODF bit is cleared to 0 under the following conditions:

- After reading SPSR in a condition where the MODF bit has turned 1, the CPU writes a 0 to the MODF bit.
- System reset

#### 17.4.8 **Initializing RSPI**

If the CPU writes a 0 to the SPE bit in the RSPI control register (SPCR) or the RSPI clears the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes a part of the module function. During a power-on reset or standby, the RSPI initializes all of the module function. An explanation follows of initialization by the clearing of the SPE bit and initialization by power-on reset/standby.

#### **(1) Initialization by Clearing SPE Bit**

When the SPE bit in SPCR is cleared, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals only in slave mode (Hi-Z)
- Initializing the internal state of the RSPI
- Initializing the SPTEF bit in the RSPI status register (SPSR)

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPI. For this reason, the RSPI can be started in the same transfer mode as prior to the initialization if the CPU resets the value 1 to the SPE bit.

The SPRF, OVRF, and MODF bits in SPSR are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.

The SPTEF bit in SPSR is initialized to 1. Therefore, if the SPTIE bit in SPCR is set to 1 after RSPI initialization, an RSPI transmit interrupt is generated. When the RSPI is initialized by the CPU, in order to disable any RSPI transmit interrupt, a 0 should be written to the SPTIE bit simultaneously with the writing of a 0 to the SPE bit. To disable any RSPI transmit interrupt after a mode fault error is detected, use an error handling routine to write a 0 to the SPTIE bit.

#### **(2)** Power-On Reset/Standby

The initialization by power-on reset/standby completely initializes the RSPI through the initialization of all bits for controlling the RSPI, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 17.4.8 (1), Initialization by Clearing SPE Bit.

### 17.4.9 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (see section 17.4.7, Error Detection). When operating in single-master mode, the RSPI does not detect mode fault errors whereas the RSPI running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-/multi-master modes.

### (1) Starting Serial Transfer

The RSPI updates the data in the transmission buffer when the SPTEF bit in the RSPI status register (SPSR) is 1 and when either the CPU or the A-DMAC has written data to the RSPI data register (SPDR). If the shift register is empty in a condition where the SPTEF bit has been cleared to 0 due to the writing of 0 either after the writing to SPDR from the A-DMAC or by the writing of 0 after the value 1 is read from the SPTEF bit by the CPU, the RSPI copies the data in the transmission buffer to the shift register and starts a serial transfer. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to "full", and upon termination of serial transfer, it changes the status of the shift register to "empty". The status of the shift register cannot be referenced from the CPU.

For details on the RSPI transfer format, see section 17.4.4, Transfer Format.

## (2) Terminating a Serial Transfer

Irrespective of the CPHA bit in the RSPI command register (SPCMD), the RSPI terminates the serial transfer after transmitting an RSPCK edge corresponding to the final sampling timing. If the SPRF bit in the RSPI status register (SPSR) is 0 and free space is available in the receive buffer, upon termination of serial transfer the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).

It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 in SPCMD. For details on the RSPI transfer format, see section 17.4.4, Transfer Format.

## (3) Sequence Control

The transfer format that is employed in master mode is determined by the RSPI sequence control register (SPSCR), RSPI command registers 0 to 7 (SPCMD0 to SPCMD7), the RSPI bit rate register (SPBR), the RSPI clock delay register (SPCKD), the RSPI slave select negation delay register (SSLND), and the RSPI next-access delay register (SPND).

The SPSCR register is used to determine the sequence configuration for serial transfers that are executed by a master mode RSPI. The following items are set in RSPI command registers SPCMD0 to SPCMD7: SSL output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMD0 to SPCMD7. The RSPI contains a pointer to the SPCMD that makes up the sequence. The value of this pointer can be checked by reading bits SPCP2 to SPCP0 in the RSPI sequence status register (SPSSR). When the SPE bit in the RSPI control register (SPCR) is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

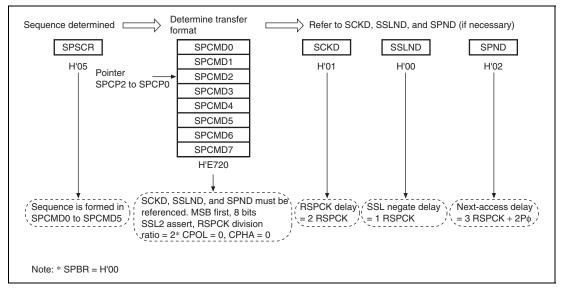


Figure 17.16 Determination Procedure of Serial Transfer Mode in Master Mode

### (4) Burst Transfer

If the SSLKP bit in the RSPI command register (SPCMD) that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSL signal level during the serial transfer until the beginning of the SSL signal assertion for the next serial transfer. If the SSL signal level for the next serial transfer is the same as the SSL signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSL signal assertion status (burst transfer).

Figure 17.17 shows an example of an SSL signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 settings. The text below explains the RSPI operations (1) to (7) as depicted in figure 17.17. It should be noted that the polarity of the SSL output signal depends on the settings in the RSPI slave select polarity register (SSLP).

- 1. Based on SPCMD0, the RSPI asserts the SSL signal and inserts RSPCK delays.
- 2. The RSPI executes serial transfers according to SPCMD0.
- 3. The RSPI inserts SSL negation delays.
- 4. Because the SSLKP bit in SPCMD0 is 1, the RSPI keeps the SSL signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until such time as the transmit data is stored in the shift register for another transfer.
- 5. Based on SPCMD1, the RSPI asserts the SSL signal and inserts RSPCK delays.
- 6. The RSPI executes serial transfers according to SPCMD1.

7. Because the SSLKP bit in SPCMD1 is 0, the RSPI negates the SSL signal. In addition, a nextaccess delay is inserted according to SPCMD1.

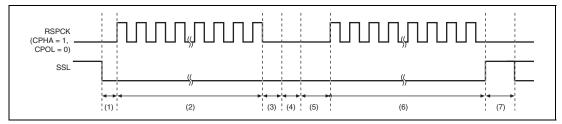


Figure 17.17 Example of Burst Transfer Operation using SSLKP Bit

If the SSL signal settings in the SPCMD in which 1 is assigned to the SSLKP bit are different from the SSL signal output settings in the SPCMD to be used in the next transfer, the RSPI switches the SSL signal status to SSL signal assertion ((5) in figure 17.17) corresponding to the command for the next transfer. Notice that if such an SSL signal switching occurs, the slaves that drive the MISO signal compete, and the possibility arises of the collision of signal levels.

The RSPI in master mode references within the module the SSL signal operation for the case where the SSLKP bit is not used. Even when the CPHA bit in SPCMD is 0, the RSPI can accurately start serial transfers by asserting the SSL signal for the next transfer. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see section 17.4.10, Slave Mode Operation).

#### (5)RSPCK Delay (t1)

The RSPCK delay value of the RSPI in master mode depends on SCKDEN bit settings in the RSPI command register (SPCMD) and on RSPCK delay register (SPCKD) settings. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SCKDEN bit in the selected SPCMD and SPCKD, as shown in table 17.8. For a definition of RSPCK delay, see section 17.4.4, Transfer Format.

Table 17.8 Relationship among SCKDEN and SPCKD Settings and RSPCK Delay Values

| SCKDEN | SPCKD      | RSPCK Delay Value |
|--------|------------|-------------------|
| 0      | 000 to 111 | 1 RSPCK           |
| 1      | 000        | 1 RSPCK           |
|        | 001        | 2 RSPCK           |
|        | 010        | 3 RSPCK           |
|        | 011        | 4 RSPCK           |
|        | 100        | 5 RSPCK           |
|        | 101        | 6 RSPCK           |
|        | 110        | 7 RSPCK           |
|        | 111        | 8 RSPCK           |

## (6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on SLNDEN bit settings in the RSPI command register (SPCMD) and on SSL negation delay register (SSLND) settings. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SLNDEN bit in the selected SPCMD and SSLND, as shown in table 17.9. For a definition of SSL negation delay, see section 17.4.4, Transfer Format.

Table 17.9 Relationship among SLNDEN and SSLND Settings and SSL Negation Delay Values

| SLNDEN | SSLND      | SSL Negation Delay Value |
|--------|------------|--------------------------|
| 0      | 000 to 111 | 1 RSPCK                  |
| 1      | 000        | 1 RSPCK                  |
|        | 001        | 2 RSPCK                  |
|        | 010        | 3 RSPCK                  |
|        | 011        | 4 RSPCK                  |
|        | 100        | 5 RSPCK                  |
|        | 101        | 6 RSPCK                  |
|        | 110        | 7 RSPCK                  |
|        | 111        | 8 RSPCK                  |

## (7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on SPNDEN bit settings in the RSPI command register (SPCMD) and on next-access delay register (SPND) settings. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPNDEN bit in the selected SPCMD and SPND, as shown in table 17.10. For a definition of next-access delay, see section 17.4.4, Transfer Format

Table 17.10 Relationship among SPNDEN and SPND Settings and Next-Access Delay Values

| SPNDEN | SPND       | Next-Access Delay Value |
|--------|------------|-------------------------|
| 0      | 000 to 111 | 1 RSPCK + 2 Pφ          |
| 1      | 000        | 1 RSPCK + 2 Pφ          |
|        | 001        | 2 RSPCK + 2 Pφ          |
|        | 010        | 3 RSPCK + 2 Pφ          |
|        | 011        | 4 RSPCK + 2 Pφ          |
|        | 100        | 5 RSPCK + 2 Pφ          |
|        | 101        | 6 RSPCK + 2 Pφ          |
|        | 110        | 7 RSPCK + 2 Pφ          |
|        | 111        | 8 RSPCK + 2 Pφ          |

### (8) Initialization Flowchart

Figure 17.18 shows an example of initialization flowchart for using the RSPI in master mode. For a description of how to set up an interrupt controller, the A-DMAC, and input/output ports, see the descriptions given in the individual blocks.

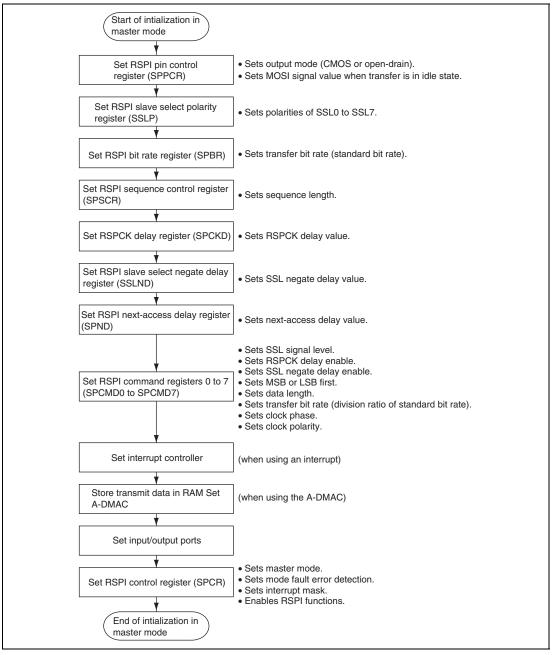


Figure 17.18 Example of Initialization Flowchart in Master Mode

#### **Transfer Operation Flowchart** (9)

Figure 17.19 shows an example of transfer operation flowchart for using the RSPI in master mode.

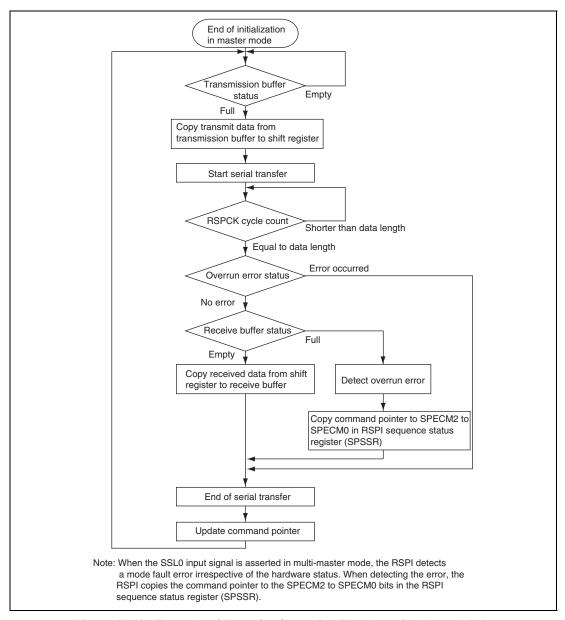


Figure 17.19 Example of Transfer Operation Flowchart in Master Mode

### 17.4.10 Slave Mode Operation

## (1) Starting a Serial Transfer

If the CPHA bit in RSPI command register 0 (SPCMD0) is 0, when detecting an SSL0 input signal assertion, the RSPI needs to start driving valid data to the MISO output signal. For this reason, when the CPHA bit is 0, the asserting of the SSL0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCK edge in an SSL0 signal asserted condition, the RSPI needs to start driving valid data to the MSO signal. For this reason, when the CPHA bit is 1, the first RSPCK edge in an SSL0 signal asserted condition triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to "full", so that data cannot be copied from the transmission buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register intact, in the full state.

Irrespective of CPHA bit settings, the timing at which the RSPI starts driving MISO output signals is the SSL0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on CPHA bit settings.

For details on the RSPI transfer format, see section 17.4.4, Transfer Format. The polarity of the SSL0 input signal depends on the setting of the SSL0P bit in the RSPI slave select polarity register (SSLP).

## (2) Terminating a Serial Transfer

Irrespective of the CPHA bit in RSPI command register 0 (SPCMD0), the RSPI terminates the serial transfer after detecting an RSPCK edge corresponding to the final sampling timing. When the SPRF bit in the RSPI status register (SPSR) is 0 and free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR). Irrespective of the value of the SPRF bit, upon termination of a serial transfer the RSPI changes the status of the shift register to "empty". When the MODFEN bit in the RSPI control register (SPCR) is 1, a mode fault error occurs if the RSPI detects an SSL0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 17.4.7, Error Detection).

The final sampling timing changes depending on the bit length of the transfer data. In slave mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 bits in SPCMD0. The polarity of the SSL0 input signal depends on the setting in the SSL0P bit in the RSPI slave select polarity register (SSLP). For details on the RSPI transfer format, see section 17.4.4, Transfer Format.

### (3) Notes on Single-Slave Operations

If the CPHA bit in RSPI command register 0(SPCMD0) is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSL0 input signal. In the type of configuration shown in figure 17.4 as an example, if the RSPI is used in single-slave mode, the SSL0 signal is always fixed at active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute send/receive operation by the RSPI in a configuration in which the SSL0 input signal is fixed at active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSL0 input signal should not be fixed.

### (4) Burst Transfer

If the CPHA bit in RSPI command register 0 (SPCMD0) is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSL0 input signal. If the CPHA bit is 1, the period from the first RSPCK edge to the sampling timing for the reception of the final bit in an SSL0 signal active state corresponds to a serial transfer period. Even when the SSL0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of access.

If the CPHA bit is 0, for the reason given in section 17.4.10 (3), Notes on Single-Slave Operations, second and subsequent serial transfers during the burst transfer cannot be executed correctly.

## (5) Initialization Flowchart

Figure 17.20 shows an example of initialization flowchart for using the RSPI in slave mode. For a description of how to set up an interrupt controller, the A-DMAC, and input/output ports, see the descriptions given in the individual blocks.

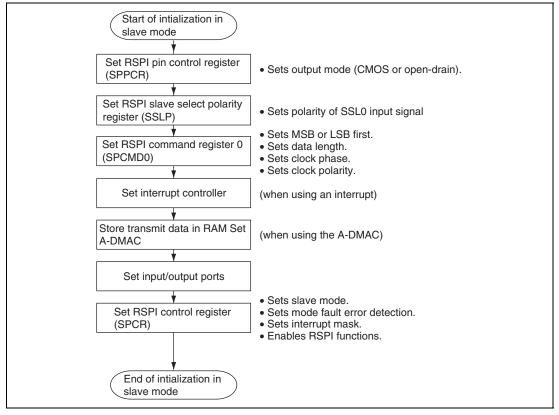


Figure 17.20 Example of Initialization Flowchart in Slave Mode

## (6) Transfer Operation Flowchart (CPHA = 0)

Figure 17.21 shows an example of transfer operation flowchart for using the RSPI in slave mode, when the CPHA bit in RSPI command register 0 (SPCMD0) is 0 and the MODFEN bit in the RSPI control register (SPCR) is 1. When the serial transfer starts while the MODFEN bit is 0, and the SSL0 input level is negated while the RSPCK cycle count is shorter than the data length, operation cannot be guaranteed.

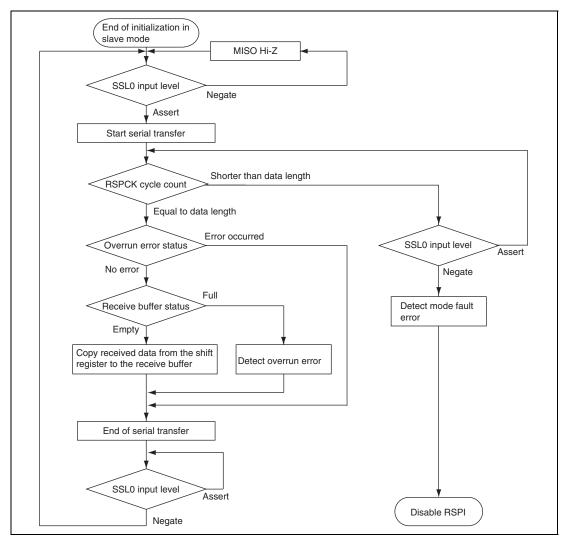


Figure 17.21 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 0, MODFEN = 1)

## (7) Transfer Operation Flowchart (CPHA = 1)

Figure 17.22 shows an example of transfer operation flowchart for using the RSPI in slave mode, when the CPHA bit in RSPI command register 0 (SPCMD0) is 1 and the MODFEN bit in the RSPI control register (SPCR) is 1. When the serial transfer starts while the MODFEN bit is 0, and the SSL0 input level is negated while the RSPCK cycle count is shorter than the data length, operation cannot be guaranteed.

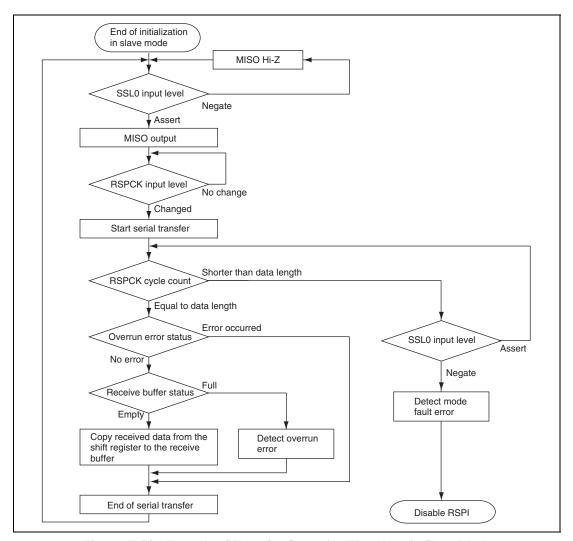


Figure 17.22 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 1, MODFEN = 1)

### 17.4.11 Loopback Mode

When the CPU writes 1 to the SPLP bit in the RSPI pin control register (SPPCR), the RSPI shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects the input path and the output path (reversed) of the shift register. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI becomes the received data for the RSPI. Figure 17.23 shows the configuration of the shift register input/output paths for the case where the RSPI in master mode is set in loopback mode.

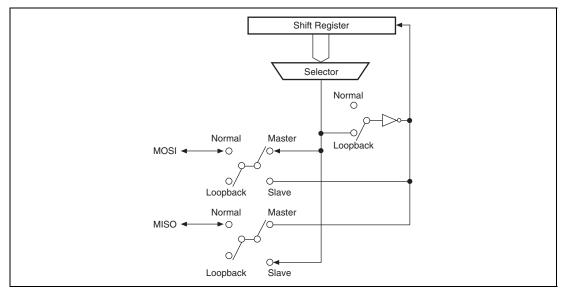


Figure 17.23 Configuration of Shift Register Input/Output Paths in Loopback Mode (Master Mode)

# Section 18 Controller Area Network (RCAN-TL1)

#### 18.1 Summary

#### 18.1.1 Overview

This document primarily describes the programming interface for the RCAN-TL1 (Renesas CAN Time Trigger Level 1) module. It serves to facilitate the hardware/software interface so that engineers involved in the RCAN-TL1 implementation can ensure the design is successful.

#### 18.1.2 Scope

The CAN Data Link Controller function is not described in this document. It is the responsibility of the reader to investigate the CAN Specification Document (see references). The interfaces from the CAN Controller are described, in so far as they pertain to the connection with the User Interface.

The programming model is described in some detail. It is not the intention of this document to describe the implementation of the programming interface, but to simply present the interface to the underlying CAN functionality.

The document places no constraints upon the implementation of the RCAN-TL1 module in terms of process, packaging or power supply criteria. These issues are resolved where appropriate in implementation specifications.

#### 18.1.3 Audience

In particular this document provides the design reference for software authors who are responsible for creating a CAN application using this module.

In the creation of the RCAN-TL1 user interface LSI engineers must use this document to understand the hardware requirements.

#### 18.1.4 References

- 1. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
- 2. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
- 3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997

- 4. Road vehicles Controller area network (CAN): Part 1: Data link layer and physical signalling (ISO-11898-1, 2003)
- Road vehicles Controller area network (CAN): Part 4: Time triggered communication (ISO-11898-4, 2004)

### 18.1.5 Features

- Supports CAN specification 2.0B
- Bit timing compliant with ISO-11898-1
- 32 Mailbox version
- Clock: 16 to 20 MHz, 32 to 40 MHz
- 31 programmable Mailboxes for transmit / receive + 1 receive-only mailbox
- Sleep mode for low power consumption and automatic recovery from sleep mode by detecting CAN bus activity
- Programmable receive filter mask (standard and extended identifier) supported by all Mailboxes
- Programmable CAN data rate up to 1MBit/s
- Transmit message queuing with internal priority sorting mechanism against the problem of priority inversion for real-time applications
- Data buffer access without SW handshake requirement in reception
- Flexible micro-controller interface
- Flexible interrupt structure
- 16-bit free running timer with flexible clock sources and pre-scaler, 3 Timer Compare Match Registers
- 6-bit Basic Cycle Counter for Time Trigger Transmission
- Timer Compare Match Registers with interrupt generation
- Timer counter clear / set capability
- Registers for Time-Trigger: Local\_Time, Cycle\_time, Ref\_Mark, Tx\_Enable Window, Ref\_Trigger\_Offset
- Flexible TimeStamp at SOF for both transmission and reception supported
- Time-Trigger Transmission, Periodic Transmission supported (on top of Event Trigger Transmission)
- Basic Cycle value can be embedded into a CAN frame and transmitted
- Parity: One parity bit is added to every 8 bits of data and parity detection is performed in 32bit units. A flag is set when a parity error is detected.

#### **Architecture** 18.2

The RCAN-TL1 device offers a flexible and sophisticated way to organise and control CAN frames, providing the compliance to CAN2.0B Active and ISO-11898-1. The module is formed from 7 different functional entities. These are the Micro Processor Interface (MPI), Mailbox, Mailbox Control, Timer, CAN Interface, Parity Control, and Parity Circuit. The figure below shows the block diagram of the RCAN-TL1 Module. The bus interface timing is designed according to the peripheral bus I/F required for each product.

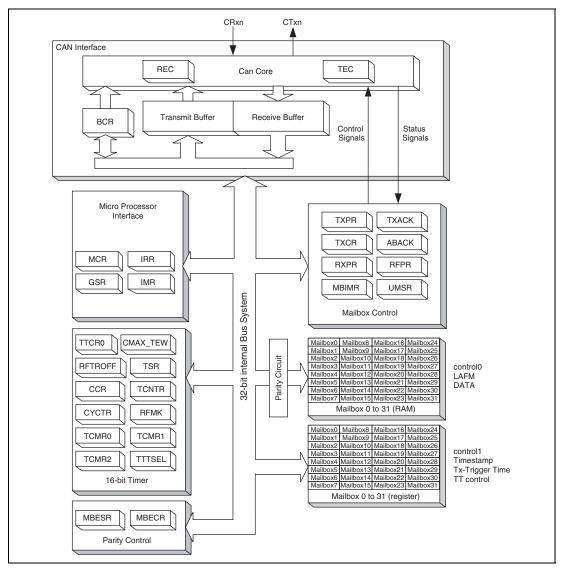


Figure 18.1 RCAN-TL1 architecture

**Important:** Although core of RCAN-TL1 is designed based on a 32-bit bus system, the whole RCAN-TL1 including MPI for the CPU has 16-bit bus interface to CPU. In that case, LongWord (32-bit) access must be implemented as 2 consecutive word (16-bit) accesses. In this manual, LongWord access means the two consecutive accesses.

### Micro Processor Interface (MPI)

The MPI allows communication between the Renesas CPU and the RCAN-TL1's registers/mailboxes to control the memory interface. It also contains the Wakeup Control logic that detects the CAN bus activities and notifies the MPI and the other parts of RCAN-TL1 so that the RCAN-TL1 can automatically exit the Sleep mode.

It contains registers such as MCR, IRR, GSR and IMR.

### Mailbox

The Mailboxes consists of RAM configured as message buffers and registers. There are 32 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)
- Time Stamp for message reception/transmission
- 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit
- Tx-Trigger Time

### Mailbox Control

The Mailbox Control handles the following functions.

- For received messages, compare the IDs and generate appropriate RAM addresses/data to store messages from the CAN Interface into the Mailbox and set/clear appropriate registers accordingly.
- To transmit event-triggered messages, run the internal arbitration to pick the correct priority message, and load the message from the Mailbox into the Tx-buffer of the CAN Interface and set/clear appropriate registers accordingly. In the case of time-triggered transmission, compare match of Tx-Trigger time invoke loading the messages.
- Arbitrates Mailbox accesses between the CPU and the Mailbox Control.
- Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, UMSR and MBIMR.

### Timer

The Timer function is the functional entity, which provides RCAN-TL1 with support for transmitting messages at a specific time frame and recording the result.

The Timer is a 16-bit free running up counter which can be controlled by the CPU. It provides one 16-bit Compare Match Register to compare with Local Time and two 16-bit ones to compare with Cycle Time. The Compare Match Registers can generate interrupt signals and clear the Counter.

The clock period of this Timer offers a wide selection derived from the system clock or can be programmed to be incremented with one nominal bit timing of CAN Bus.

Contains registers such as TCNTR, TTCR0, CMAX\_TEW, RETROFF, TSR, CCR, CYCTR, RFMK, TCMR0, TCMR1, TCMR2 and TTTSEL.

### CAN Interface

This block conforms to the requirements for a CAN Bus Data Link Controller which is specified in Ref. [2, 4]. It fulfils all the functions of a standard DLC as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and the logic which are specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Configuration Registers and various useful Test Modes. This block also contains functional entities to hold the data received and the data to be transmitted for the CAN Data Link Controller.

## Parity Control

The Parity Control block allows communications between the CPU and the two parity control registers, MBESR and MBECR.

## Parity Circuit

Parity bits are added to the data written to a mailbox (RAM). One parity bit is added to every 8 bits of the data.

Parity check is performed to the data read from a mailbox (RAM), in 32-bit units.

All mailboxes (RAM) must be initialized. If any data is read from a mailbox (RAM) without initialization, a parity error may occur.

Generation of a parity error interrupt can be selected by setting the MBECR register.

# 18.3 Programming Model - Overview

The purpose of this programming interface is to allow convenient, effective access to the CAN bus for efficient message transfer. Please bear in mind that the user manual reports all settings allowed by the RCAN-TL1 IP. Different use of RCAN-TL1 is not allowed.

## 18.3.1 Memory Map

The diagram of the memory map is shown below.

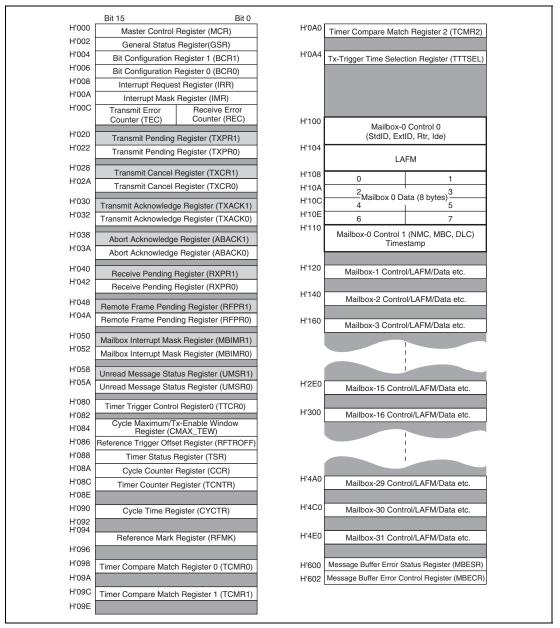


Figure 18.2 RCAN-TL1 Memory Map

The locations not used (between H'000 and H'602) are reserved and cannot be accessed.

### 18.3.2 Mailbox Structure

Mailboxes play a role as message buffers to transmit/receive CAN frames. Each Mailbox is comprised of 3 identical storage fields that are 1): Message Control, 2): Local Acceptance Filter Mask, 3): Message Data. In addition some Mailboxes contain the following extra Fields: 4): Time Stamp, 5): Time Trigger configuration and 6): Time Trigger Control. The following table shows the address map for the control, LAFM, data, timestamp, Transmission Trigger Time and Time Trigger Control addresses for each mailbox.

### Address

|          | Control0      | LAFM          | Data          | Control1      | Time Stamp    | Trigger Time | TT control |
|----------|---------------|---------------|---------------|---------------|---------------|--------------|------------|
| Mailbox  | 4 bytes       | 4 bytes       | 8 bytes       | 2 bytes       | 2 bytes       | 2 bytes      | 2 bytes    |
| 0        |               |               |               |               |               |              |            |
| (Receive | H'100 – H'103 | H'104- H'107  | H'108 – H'10F | H'110 – H'111 | H'112 – H'113 | No           | No         |
| Only)    |               |               |               |               |               |              |            |
| 1        | H'120 – H'123 | H'124 – H'127 | H'128 – H'12F | H'130 – H'131 | H'132 – H'133 | No           | No         |
| 2        | H'140 – H'143 | H'144 – H'147 | H'148 – H'14F | H'150 – H'151 | H'152 – H'153 | No           | No         |
| 3        | H'160 – H'163 | H'164 - H'167 | H'168 – H'16F | H'170 – H'171 | H'172 – H'173 | No           | No         |
| 4        | H'180 – H'183 | H'184 – H'187 | H'188 – H'18F | H'190 – H'191 | H'192 – H'193 | No           | No         |
| 5        | H'1A0 – H'1A3 | H'1A4 – H'1A7 | H'1A8 – H'1AF | H'1B0 – H'1B1 | H'1B2 – H'1B3 | No           | No         |
| 6        | H'1C0 - H'1C3 | H'1C4 – H'1C7 | H'1C8 – H'1CF | H'1D0 – H'1D1 | H'1D2 – H'1D3 | No           | No         |
| 7        | H'1E0 – H'1E3 | H'1E4 – H'1E7 | H'1E8 – H'1EF | H'1F0 – H'1F1 | H'1F2 – H'1F3 | No           | No         |
| 8        | H'200 – H'203 | H'204 – H'207 | H'208 – H'20F | H'210 – H'211 | H'212 – H'213 | No           | No         |
| 9        | H'220 – H'223 | H'224 – H'227 | H'228 – H'22F | H'230 – H'231 | H'232 – H'233 | No           | No         |
| 10       | H'240 – H'243 | H'244 – H'247 | H'248 – H'24F | H'250 – H'251 | H'252 – H'253 | No           | No         |
| 11       | H'260 – H'263 | H'264 – H'267 | H'268 – H'26F | H'270 – H'271 | H'272 – H'273 | No           | No         |
| 12       | H'280 – H'283 | H'284 – H'287 | H'288 – H'28F | H'290 – H'291 | H'292 – H'293 | No           | No         |
| 13       | H'2A0 – H'2A3 | H'2A4 – H'2A7 | H'2A8 – H'2AF | H'2B0 – H'2B1 | H'2B2 – H'2B3 | No           | No         |
| 14       | H'2C0 – H'2C3 | H'2C4 – H'2C7 | H'2C8 – H'2CF | H'2D0 – H'2D1 | H'2D2 – H'2D3 | No           | No         |
| 15       | H'2E0 – H'2E3 | H'2E4 – H'2E7 | H'2E8 – H'2EF | H'2F0 – H'2F1 | H'2F2 – H'2F3 | No           | No         |
| 16       | H'300 – H'303 | H'304 – H'307 | H'308 – H'30F | H'310 – H'311 | No            | No           | No         |
| 17       | H'320 – H'323 | H'324 – H'327 | H'328 – H'32F | H'330 – H'331 | No            | No           | No         |
| 18       | H'340 – H'343 | H'344 – H'347 | H'348 – H'34F | H'350 – H'351 | No            | No           | No         |

### Address

|         | Control0      | LAFM          | Data          | Control1      | Time Stamp    | Trigger Time  | TT control    |
|---------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Mailbox | 4 bytes       | 4 bytes       | 8 bytes       | 2 bytes       | 2 bytes       | 2 bytes       | 2 bytes       |
| 19      | H'360 – H'363 | H'364 – H'367 | H'368 – H'36F | H'370 – H'371 | No            | No            | No            |
| 20      | H'380 – H'383 | H'384 – H'387 | H'388 – H'38F | H'390 – H'391 | No            | No            | No            |
| 21      | H'3A0 – H'3A3 | H'3A4 – H'3A7 | H'3A8 – H'3AF | H'3B0 – H'3B1 | No            | No            | No            |
| 22      | H'3C0 - H'3C3 | H'3C4 – H'3C7 | H'3C8 – H'3CF | H'3D0 – H'3D1 | No            | No            | No            |
| 23      | H'3E0 – H'3E3 | H'3E4 – H'3E7 | H'3E8 – H'3EF | H'3F0 – H'3F1 | No            | No            | No            |
| 24      | H'400 – H'403 | H'404 – H'407 | H'408 – H'40F | H'410 – H'411 | No            | H'414 – H'415 | H'416 – H'417 |
| 25      | H'420 – H'423 | H'424 – H'427 | H'428 – H'42F | H'430 – H'431 | No            | H'434 – H'435 | H'436 – H'437 |
| 26      | H'440 – H'443 | H'444 – H'447 | H'448 – H'44F | H'450 – H'451 | No            | H'454 – H'455 | H'456 – H'457 |
| 27      | H'460 – H'463 | H'464 – H'467 | H'468 – H'46F | H'470 – H'471 | No            | H'474 – H'475 | H'476 – H'477 |
| 28      | H'480 – H'483 | H'484 – H'487 | H'488 – H'48F | H'490 – H'491 | No            | H'494 – H'495 | H'496 – H'497 |
| 29      | H'4A0 – H'4A3 | H'4A4 – H'4A7 | H'4A8 – H'4AF | H'4B0 – H'4B1 | No            | H'4B4 – H'4B5 | H'4B6 – H'4B7 |
| 30      | 111400 111400 | H'4C4 – H'4C7 | 111400 111405 | LIMPO LIMPA   | H'4D2 – H'4D3 | H'4D4 – H'4D5 | No            |
| 30      | п 400 – п 403 | П 404 — П 407 | П 4C6 – П 4CF | H 4D0 - H 4D1 | (Local Time)  | H 4D4 – H 4D5 | INO           |
| 31      | שיאבט שיאבט   | H'4E4 – H'4E7 | UNEO   UNEE   | HIVEO HIVE1   | H'4F2 – H'4F3 | No            | No            |
| 31      | 11460 - 11463 | 11464 - 1146/ | 11460 - 11466 | 11470-11471   | (Local Time)  | INU           | INU           |

Mailbox-0 is a receive-only box, and all the other Mailboxes can operate as both receive and transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message Control. The following diagram shows the structure of a Mailbox in detail.

**Table 18.1 Roles of Mailboxes** 

|           | Even | t Trigger | Time T  | rigger                          | Rei       | mark               |
|-----------|------|-----------|---|---------------------------------|-----------|--------------------|
|           | Тх   | Rx        | Тх  | Rx                              | TimeStamp | Tx-Trigger<br>Time |
| MB31      | OK   | OK        | _   | time reference reception        | available | _                  |
| MB30      | OK   | OK        | time reference<br>transmission in time<br>master mode | reception in time<br>slave mode | available | available          |
| MB29 - 24 | OK   | OK        | Setting available                                     | Setting available               | _         | available          |
| MB23 - 16 | OK   | OK        | — (ET)  | Setting available               | _         | _                  |
| MB15 - 1  | OK   | OK        | — (ET)  | Setting available               | available | _                  |
| MB0       | _    | OK        | _   | Setting available               | available |                    |

(ET) shows that it works during merged arbitrating window, after completion of time-triggered transmission.

|   |   |           |                        |  |                      |         |           | ata Bus                     |  |        |                   |                       |                |          |               |   |                                   |
|---|---|-----------|------------------------|--|----------------------|---------|-----------|-----------------------------|--|--------|-------------------|-----------------------|----------------|----------|---------------|---|-----------------------------------|
| Address   | 15  | 14        | 13                     | 12   | 11                   | 10      | 9         | 3 7                         | 6                                      | 5      | 4                 | 3                     | 2              | 1        | 0             | Access Size   | Field Name                        |
| H'100   | IDE   | RTR       | 0                      |  |                      |         |           | STD                         | ID[10:0]                               |        |                   |                       |                | EXTI     | 0[17:16]      | Word/LW   | Control 0                         |
| H'102   |   |           |                        |  |                      |         | E.        | KTID[15:0]                  | ]                                      |        |                   |                       |                | •        |               | Word  | Control 0                         |
| H'104   | IDE LAFM         0         0         STDID_LAFM[10:0]         EXTID LAFM[17:16] |           |                        |  |                      |         |           |                             |  |        | Word/LW           | LAFM                  |                |          |               |   |                                   |
| H'106   | EXTID_LAFM[15:0]  |           |                        |  |                      |         |           |                             |  |        |                   | Word                  | LAFIVI         |          |               |   |                                   |
| H'108   |   | MS        | G_DAT                  | ΓA_0 (fi   | irst Rx/             | Tx Byte | )         |                             |  |        | MSG               | _DATA_                | 1              |          |               | Byte/Word/LW  |                                   |
| H'10A   |   |           | N                      | ISG_D/   | ATA_2                |         |           |                             |  |        | MSG               | _DATA_                | 3              |          |               | Byte/Word   | Data                              |
| H'10C   |   |           | M                      | ISG_D/   | ATA_4                |         |           |                             |  |        | MSG               | DATA                  | 5              |          |               | Byte/Word/LW  | Data                              |
| H'10E   |   |           | N                      | ISG_D/   | ATA_6                |         |           |                             |  |        | MSG               | _DATA_                | 7              |          |               | Byte/Word   |                                   |
| H'110   | 0   | 0         | NMC                    | 0  | 0                    |         | MBC[2:0]  | 0                           | 0                                      | 0      | 0                 |                       | DLC            | [3:0]    |               | Byte/Word   | Control 1                         |
| H'112   |   |           |                        |  |                      |         |           |                             |  |        |                   |                       |                |          |               |   |                                   |
|   | with tim  | nestamp   |                        | TimeS  |                      | - ' '   | YCTR[15:0 |                             | 5:0]/CY0                               | CTR[15 | 5:6] at S         | OF)                   |                |          |               | Word  | TimeStamp                         |
| B15 to 1 (MB  |   |           | )                      | SC[1] is   | fixed to             | ) "1"   |           | Data Bus                    |  |        |                   |                       |                | 1        |               |   |                                   |
| B15 to 1 (MB  | 15  | 14        | 13                     |  |                      | - ' '   |           |                             | 6                                      | 5 ETR  | 5:6] at S         | 3                     | 2              | 1        | 0             | Access Size   | TimeStamp Field Name              |
| B15 to 1 (MB<br>Address<br>d'100 + n*32   |   |           | )                      | SC[1] is   | fixed to             | ) "1"   | 9   1     | Data Bus<br>3 7<br>STD      | 6<br>ID[10:0]                          | 5      |                   |                       | 2              | <u> </u> | 0<br>D[17:16] | Access Size Word/LW   |                                   |
| B15 to 1 (MB<br>Address<br>H'100 + n*32<br>H'102 + n*32   | 15<br>IDE   | 14<br>RTR | 13                     | SC[1] is   | fixed to             | ) "1"   | 9   1     | Data Bus 3 7 STD KTID[15:0] | 6<br>ID[10:0]                          | 5      |                   |                       | 2              | EXTI     | 0[17:16]      | Access Size Word/LW Word  | Field Name                        |
| Address 4100 + n*32 4102 + n*32 4104 + n*32   | 15  | 14<br>RTR | 13                     | SC[1] is   | fixed to             | ) "1"   | 9   E     | Data Bus 3 7 STD  STDID_    | 6<br>   6<br> D[10:0]<br> <br>  LAFM[1 | 5      |                   |                       | 2              | EXTI     | -             | Access Size Word/LW Word Word/LW  | Field Name                        |
| Address d'100 + n*32 d'102 + n*32 d'104 + n*32 d'106 + n*32   | 15<br>IDE   | 14<br>RTR | 13                     | SC[1] is   | fixed to             | ) "1"   | 9   E     | Data Bus 3 7 STD KTID[15:0] | 6<br>   6<br> D[10:0]<br> <br>  LAFM[1 | 5      |                   |                       | 2              | EXTI     | D[17:16]      | Access Size Word/LW Word Word/LW Word   | Field Name<br>Control 0           |
| Address   | 15<br>IDE   | 14<br>RTR | 13 0                   | 12   | fixed to             | ) "1"   | 9 E       | Data Bus 3 7 STD  STDID_    | 6<br>   6<br> D[10:0]<br> <br>  LAFM[1 | 5      | 4                 |                       |                | EXTI     | D[17:16]      | Access Size Word/LW Word Word/LW Word Byte/Word/LW                                  | Field Name<br>Control 0           |
| Address  1'100 + n*32  1'102 + n*32  1'104 + n*32  1'106 + n*32  1'108 + n*32  1'10A + n*32               | 15<br>IDE   | 14<br>RTR | 13<br>0<br>0           | 12   | fixed to             | 10      | 9 E       | Data Bus 3 7 STD  STDID_    | 6<br>   6<br> D[10:0]<br> <br>  LAFM[1 | 5      | 4 MSG             | 3                     | .1             | EXTI     | D[17:16]      | Access Size Word/LW Word Word/LW Word Byte/Word/LW Byte/Word                        | Field Name Control 0  LAFM        |
| Address  1'100 + n*32 1'102 + n*32 1'104 + n*32 1'106 + n*32 1'108 + n*32 1'10A + n*32 1'10C + n*32       | 15<br>IDE   | 14<br>RTR | 13<br>0<br>0<br>GG_DA1 | 12<br>TA_0 (fi   | fixed to             | 10      | 9 E       | Data Bus 3 7 STD  STDID_    | 6<br>   6<br> D[10:0]<br> <br>  LAFM[1 | 5      | MSG<br>MSG        | 3<br>_DATA_           | .1             | EXTI     | D[17:16]      | Access Size Word/LW Word Word/LW Word Byte/Word/LW                                  | Field Name<br>Control 0           |
| Address  d'100 + n*32 d'102 + n*32 d'104 + n*32 d'106 + n*32 d'108 + n*32                                 | 15<br>IDE   | 14<br>RTR | 13<br>0<br>0<br>G_DAT  | 12 TA_0 (fi  | 11  11  ATA_2  ATA_4 | 10      | 9 E       | Data Bus 3 7 STD  STDID_    | 6<br>   6<br> D[10:0]<br> <br>  LAFM[1 | 5      | MSG<br>MSG<br>MSG | 3<br>_DATA_<br>_DATA_ | .1<br>.3<br>.5 | EXTI     | D[17:16]      | Access Size Word/LW Word Word/LW Word Byte/Word/LW Byte/Word                        | Field Name  Control 0  LAFM  Data |
| Address  4'100 + n*32  4'102 + n*32  4'104 + n*32  4'106 + n*32  4'108 + n*32  4'104 + n*32  4'107 + n*32 | 15<br>IDE   | 14<br>RTR | 13<br>0<br>0<br>G_DAT  | 12  TA_0 (fill ISG_D/ISG | 11  11  ATA_2  ATA_4 | 10      | 9 E       | Data Bus 3 7 STD  STDID_    | 6<br>   6<br> D[10:0]<br> <br>  LAFM[1 | 5      | MSG<br>MSG<br>MSG | JATA_DATA_DATA_DATA   | .1<br>.3<br>.5 | EXTI     | D[17:16]      | Access Size Word/LW Word Word/LW Word Byte/Word/LW Byte/Word Byte/Word Byte/Word/LW | Field Name Control 0  LAFM        |

Figure 18.3 Mailbox-n Structure

Legend: n = 16 to 29

|  |                          |           |                             |                          |                                  |          |              | Data                       | Bus                                       |                     |      |                |            |                      |                |              |              |   |                                  |
|--|--------------------------|-----------|-----------------------------|--------------------------|----------------------------------|----------|--------------|----------------------------|---|---------------------|------|----------------|------------|----------------------|----------------|--------------|--------------|---|----------------------------------|
| Address  | 15                       | 14        | 13                          | 12                       | 11                               | 10       | 9            | 8                          | 7   | 6                   | 5    | 4              |            | 3                    | 2              | 1            | 0            | Access Size   | Field Name                       |
| H'100 + n*32   | IDE                      | RTR       | 0                           |                          |                                  |          |              |                            | STDII                                     | D[10:0]             |      |                |            |                      |                | EXTID        | 0[17:16]     | Word/LW   | 0                                |
| H'102 + n*32   |                          |           |                             |                          |                                  |          |              | EXTI                       | 0[15:0]                                   |                     |      |                |            |                      |                | •            |              | Word  | Control 0                        |
| H'104 + n*32   | LAFM[17:16]              |           |                             |                          |                                  |          |              |                            |   |                     |      | Word/LW        | LAFM       |                      |                |              |              |   |                                  |
| H'106 + n*32   |                          |           |                             |                          |                                  |          |              |                            |   |                     |      | Word           | LAFIVI     |                      |                |              |              |   |                                  |
| H'108 + n*32   |                          | MS        | G_DAT                       | A_0 (fi                  | rst Rx/1                         | x Byte   | )            |                            |   |                     |      | MS             | SG_        | DATA                 | _1             |              |              | Byte/Word/LW  |                                  |
| H'10A + n*32   |                          |           | М                           | SG_D/                    | ATA_2                            |          |              |                            |   |                     |      | MS             | SG_        | DATA                 | _3             |              |              | Byte/Word   | Data                             |
| H'10C + n*32   |                          |           | М                           | SG_D/                    | ATA_4                            |          |              |                            |   |                     |      | MS             | SG_        | DATA                 | _5             |              |              | Byte/Word/LW  | Data                             |
| H'10E + n*32   |                          |           | М                           | SG_D/                    | ATA_6                            |          |              |                            |   |                     |      | MS             | SG_        | DATA                 | _7             |              |              | Byte/Word   |                                  |
| H'110 + n*32   | 0                        | 0         | NMC                         | ATX                      | DART                             |          | MBC[2:       | 0]                         | 0   | 0                   | 0    | 0              | Т          |                      | DLC[           | 3:0]         |              | Byte/Word   | Control 1                        |
| MB29 to 24 (   | Time-Tı                  | riggered  | d Transr                    | mission                  | in Time                          | e Trigg  | er mode      | e)<br>Data                 | Bus                                       |                     |      |                |            |                      |                |              |              |   |                                  |
| MB29 to 24 (   | Time-Ti                  | riggered  | Transi                      | mission<br>12            | in Time                          | e Trigge | er mode      |                            | Bus<br>7                                  | 6                   | 5    | 4              |            | 3                    | 2              | 1            | 0            | Access Size   | Field Name                       |
| Address  |                          |           |                             |                          |                                  |          |              | Data                       | 7   | 6<br>D[10:0]        | 5    | 4              | _          | 3                    | 2              | <del>-</del> | 0<br>[17:16] | Access Size   |                                  |
| Address<br>H'100 + n*32  | 15                       | 14        | 13                          |                          |                                  |          |              | Data<br>8                  | 7   |                     | 5    | 4              | I          | 3                    | 2              | EXTIE        | 0[17:16]     |   | Field Name Control 0             |
| Address<br>H'100 + n*32<br>H'102 + n*32  | 15                       | 14<br>RTR | 13                          |                          |                                  |          |              | Data<br>8                  | 7<br>STDII<br>D[15:0]                     |                     |      | 4              | _          | 3                    | 2              | EXTID        | 0[17:16]     | Word/LW   | Control 0                        |
| Address<br>H'100 + n*32<br>H'102 + n*32<br>H'104 + n*32  | 15<br>IDE                | 14<br>RTR | 13                          |                          |                                  |          | 9            | Data<br>8                  | 7<br>STDII<br>D[15:0]                     | D[10:0]<br>.AFM[10  |      | 4              |            | 3                    | 2              | EXTID        | D[17:16]     | Word/LW<br>Word   |                                  |
| Address<br>H'100 + n*32<br>H'102 + n*32<br>H'104 + n*32<br>H'106 + n*32  | 15<br>IDE                | 14<br>RTR | 13 0                        | 12                       |                                  | 10       | 9<br>E       | Data<br>8<br>EXTII         | 7<br>STDII<br>D[15:0]                     | D[10:0]<br>.AFM[10  |      |                |            | 3                    |                | EXTID        | D[17:16]     | Word/LW<br>Word<br>Word/LW  | Control 0                        |
| Address H'100 + n*32 H'102 + n*32 H'104 + n*32 H'106 + n*32 H'108 + n*32   | 15<br>IDE                | 14<br>RTR | 13<br>0<br>0                | 12                       | 11                               | 10       | 9<br>E       | Data<br>8<br>EXTII         | 7<br>STDII<br>D[15:0]                     | D[10:0]<br>.AFM[10  |      | MS             | _          |                      | _1             | EXTID        | D[17:16]     | Word/LW<br>Word<br>Word/LW<br>Word  | Control 0                        |
| Address H'100 + n*32 H'102 + n*32 H'104 + n*32 H'106 + n*32 H'108 + n*32 H'10A + n*32  | 15<br>IDE                | 14<br>RTR | 13<br>0<br>0<br>G_DAT       | 12<br>                   | 11 rst Rx/I                      | 10       | 9<br>E       | Data<br>8<br>EXTII         | 7<br>STDII<br>D[15:0]                     | D[10:0]<br>.AFM[10  |      | MS             | G_         | DATA                 | _1             | EXTID        | D[17:16]     | Word/LW Word Word/LW Word Byte/Word/LW  | Control 0                        |
| Address H'100 + n*32 H'102 + n*32 H'104 + n*32 H'106 + n*32 H'108 + n*32 H'108 + n*32 H'10A + n*32                           | 15<br>IDE                | 14<br>RTR | 13<br>0<br>0<br>6G_DAT<br>M | 12<br>FA_0 (fi           | 11 rst Rx/I                      | 10       | 9<br>E       | Data<br>8<br>EXTII         | 7<br>STDII<br>D[15:0]                     | D[10:0]<br>.AFM[10  |      | MS<br>MS       | SG_<br>SG_ | _DATA_               | _1<br>_3<br>_5 | EXTID        | D[17:16]     | Word/LW Word Word/LW Word Byte/Word/LW Byte/Word  | Control 0                        |
| Address H'100 + n*32 H'102 + n*32 H'104 + n*32 H'106 + n*32 H'108 + n*32 H'10A + n*32 H'10C + n*32 H'10E + n*32 H'10E + n*32 | 15<br>IDE                | 14<br>RTR | 13<br>0<br>0<br>G_DAT<br>M  | 12  FA_0 (fi SG_DA SG_DA | 11 rst Rx/I                      | 10       | 9<br>E       | Data  8  EXTIL  S'  XTID_L | 7<br>STDII<br>D[15:0]                     | D[10:0]<br>.AFM[10  |      | MS<br>MS       | SG_<br>SG_ | DATA<br>DATA<br>DATA | _1<br>_3<br>_5 | EXTID        | D[17:16]     | Word/LW Word Word/LW Word Byte/Word/LW Byte/Word Byte/Word/LW                           | Control 0                        |
| Address H'100 + n*32 H'102 + n*32 H'104 + n*32 H'106 + n*32 H'108 + n*32 H'10A + n*32 H'10C + n*32 H'10E + n*32 H'110 + n*32 | 15<br>IDE<br>IDE<br>LAFM | 14 RTR    | 13<br>0<br>0<br>G_DAT<br>M  | 12  FA_0 (fi SG_DA SG_DA | 11<br>rst Rx/I<br>ATA_2<br>ATA_4 | 10       | 9<br>E.      | Data  8  EXTII  S'  XTID_L | 7<br>STDII<br>D[15:0]<br>TDID_L<br>AFM[15 | D[10:0] AFM[10:5:0] | 0:0] | MS<br>MS<br>MS | SG_<br>SG_ | DATA<br>DATA<br>DATA | _1<br>_3<br>_5 | EXTID        | D[17:16]     | Word/LW Word Word/LW Word Byte/Word/LW Byte/Word/LW Byte/Word/LW Byte/Word/LW           | Control 0  LAFM  Data  Control 1 |
|  | 15<br>IDE<br>IDE<br>LAFM | 14 RTR    | 13<br>0<br>0<br>G_DAT<br>M  | 12  FA_0 (fi SG_DA SG_DA | 11<br>rst Rx/I<br>ATA_2<br>ATA_4 | 10       | 9<br>E:<br>) | Data  8  EXTII  S'  XTID_L | 7 STDII D[15:0] FDID_L AFM[15  0          | D[10:0] AFM[10:5:0] | 0:0] | MS<br>MS<br>MS | SG_<br>SG_ | DATA<br>DATA<br>DATA | _1<br>_3<br>_5 | EXTID        | D[17:16]     | Word/LW Word Word/LW Word Byte/Word/LW Byte/Word Byte/Word/LW Byte/Word/LW Byte/Word/LW | Control 0  LAFM  Data            |

Figure 18.3 Mailbox-n Structure (continued)

| Address   | L.               |           |                            |               |                         |                                |         | Data                                 | Bus                        |                         |                         |                   |         |       |       |              | Access Size   | Field Name                             |
|---|------------------|-----------|----------------------------|---------------|-------------------------|--------------------------------|---------|--------------------------------------|----------------------------|-------------------------|-------------------------|-------------------|---------|-------|-------|--------------|---|--|
|   | 15               | 14        | 13                         | 12            | 11                      | 10                             | 9       | 8                                    | 7                          | 6 5                     | - 4                     | 1                 | 3       | 2     | 1     | 0            | Access cize   | riciditanic                            |
| H'100 + n*32  | IDE              | RTR       | 0                          |               |                         |                                | 5       | STDID[1                              | 10:0]                      |                         |                         |                   |         |       | EXTID | [17:16]      | Word/LW   | Control 0                              |
| H'102 + n*32  |                  |           |                            |               |                         |                                |         | EXTID                                | [15:0]                     |                         |                         |                   |         |       |       |              | Word  | 0011110110                             |
| H'104 + n*32  | EAFM[17:1        |           |                            |               |                         |                                |         |                                      |                            |                         |                         | וט_<br>[17:16]    | Word/LW | LAFM  |       |              |   |  |
| H'106 + n*32  | EXTID_LAFM[15:0] |           |                            |               |                         |                                |         |                                      |                            |                         |                         |                   | Word    | LAT W |       |              |   |  |
| H'108 + n*32  |                  | MS        | G_DA                       | ΓA_0 (f       | irst Rx/                | Tx Byte                        | )       |                                      |                            | MSG                     | _DATA                   | \_1               |         |       |       |              | Byte/Word/LW  |  |
| H'10A + n*32  |                  |           | N                          | ISG_D         | ATA_2                   |                                |         |                                      |                            | MSG                     | _DATA                   | 1_3               |         |       |       |              | Byte/Word   | Data                                   |
| H'10C + n*32  |                  |           | N                          | ISG_D         | ATA_4                   |                                |         |                                      |                            | MSG                     | _DATA                   | 1_5               |         |       |       |              | Byte/Word/LW  | Julia                                  |
| H'10E + n*32  |                  |           | N                          | ISG_D         | ATA_6                   |                                |         |                                      |                            | MSG                     | _DATA                   | 1_7               |         |       |       |              | Byte/Word   |  |
| H'110 + n*32  | 0                | 0         | NMC                        | ATX           | DART                    |                                | MBC[2   | 2:0]                                 | 0                          | 0 (                     | - (                     | )                 |         | DLC   | [3:0] |              | Byte/Word   | Control 1                              |
|   |                  |           |                            |               |                         |                                |         |                                      |                            |                         |                         |                   |         |       |       |              |   |  |
| H'112 + n*32  |                  |           |                            |               |                         | TimeStamp[15:0] (TCNTR at SOF) |         |                                      |                            |                         |                         |                   |         |       |       |              | Word  | TimeStamp                              |
| H'112 + n*32<br>H'114 + n*32<br>MB31 (Time F  | Referen          | ce Rec    | eption i                   | n Time        |                         | -Trigge                        | red Tim | . , ,                                |                            | t SOF)<br>Referen       | ce                      |                   |         |       |       |              | Word<br>Word  | TimeStamp Trigger Time                 |
| H'114 + n*32<br>MB31 (Time F  | Referen          | ce Rec    | eption i                   | n Time        |                         | -Trigge                        | red Tim | . , ,                                | ) as Time                  |                         | e                       |                   |         |       |       |              | Word  | Trigger Time                           |
| H'114 + n*32  | Referen          | ce Rec    | eption i                   | n Time        |                         | -Trigge                        | red Tim | ne (TTT                              | ) as Time                  |                         |                         | 1                 | 3       | 2     | 1     | 0            |   |  |
| H'114 + n*32<br>MB31 (Time F  |                  |           |                            |               | Trigger                 | -Trigge<br>mode)               | red Tim | Data                                 | ) as Time                  | Referen                 |                         | 1                 | 3       | 2     |       | 0<br>[17:16] | Word  | Trigger Time Field Name                |
| H'114 + n*32<br>MB31 (Time F<br>Address   | 15               | 14        | 13                         |               | Trigger                 | -Trigge<br>mode)               | red Tim | Data                                 | Bus 7 0:0]                 | Referen                 |                         | 1                 | 3       | 2     |       | -            | Word Access Size  | Trigger Time                           |
| MB31 (Time F<br>Address<br>H'100 + n*32   | 15<br>IDE        | 14<br>RTR | 13                         |               | Trigger                 | -Trigge<br>mode)               | red Tim | Data<br>8                            | Bus 7 0:0]                 | Referen                 |                         | 1                 | 3       | 2     | EXTID | [17:16]      | Word  Access Size  Word/LW  | Trigger Time Field Name Control 0      |
| MB31 (Time III<br>Address<br>H'100 + n*32<br>H'102 + n*32   | 15               | 14<br>RTR | 13                         |               | Trigger                 | -Trigge<br>mode)               | 9 ST    | Data  8 TDID[10 EXTID                | Bus 7 0:0]                 | Referen                 |                         | 1                 | 3       | 2     | EXTID | [17:16]      | Word  Access Size  Word/LW  Word  | Trigger Time Field Name                |
| MB31 (Time III<br>Address<br>H'100 + n*32<br>H'102 + n*32<br>H'104 + n*32   | 15<br>IDE        | 14<br>RTR | 13 0                       | 12            | Trigger                 | -Trigge<br>mode)               | 9 ST    | Data  8 TDID[10 EXTID                | Bus 7 0:0] 0[15:0] 0[10:0] | Referen                 |                         |                   | 3       | 2     | EXTID | [17:16]      | Word  Access Size  Word/LW  Word  Word/LW   | Trigger Time Field Name Control 0      |
| MB31 (Time f<br>Address<br>H'100 + n*32<br>H'102 + n*32<br>H'104 + n*32<br>H'106 + n*32   | 15<br>IDE        | 14<br>RTR | 13<br>0<br>0               | 12            | Trigger                 | -Trigge<br>mode)               | 9 ST    | Data  8 TDID[10 EXTID                | Bus 7 0:0] 0[15:0] 0[10:0] | 6 5 MSG                 |                         | \_1               | 3       | 2     | EXTID | [17:16]      | Word  Access Size  Word/LW  Word  Word/LW  Word  Word/LW                                  | Trigger Time Field Name Control 0 LAFM |
| MB31 (Time I<br>Address<br>H'100 + n*32<br>H'102 + n*32<br>H'104 + n*32<br>H'106 + n*32<br>H'108 + n*32                                 | 15<br>IDE        | 14<br>RTR | 13<br>0<br>0               | 12<br>FA_0 (f | 11 irst Rx/             | -Trigge<br>mode)               | 9 ST    | Data  8 TDID[10 EXTID                | Bus 7 0:0] 0[15:0] 0[10:0] | Referen  6 5  MSG MSG   | _DAT/                   | \_1<br>\_3        | 3       | 2     | EXTID | [17:16]      | Word  Access Size  Word/LW  Word  Word/LW  Word  Byte/Word/LW                             | Trigger Time Field Name Control 0      |
| MB31 (Time F<br>Address<br>H'110+ n*32<br>H'102+ n*32<br>H'104+ n*32<br>H'106+ n*32<br>H'108+ n*32<br>H'108+ n*32                       | 15<br>IDE        | 14<br>RTR | 13<br>0<br>0<br>0<br>G_DA  | 12<br>FA_0 (f | 11 irst Rx/ ATA_2 ATA_4 | -Trigge<br>mode)               | 9 ST    | Data  8 TDID[10 EXTID                | Bus 7 0:0] 0[15:0] 0[10:0] | Beferen  6   5  MSG MSG | _DAT/                   | \_1<br>\_3<br>\_5 | 3       | 2     | EXTID | [17:16]      | Word  Access Size  Word/LW  Word  Word/LW  Word  Byte/Word/LW  Byte/Word/                 | Trigger Time Field Name Control 0 LAFM |
| MB31 (Time F<br>Address<br>H'100 + n*32<br>H'100 + n*32<br>H'104 + n*32<br>H'106 + n*32<br>H'108 + n*32<br>H'10A + n*32<br>H'10A + n*32 | 15<br>IDE        | 14<br>RTR | 13<br>0<br>0<br>6G_DA<br>M | TA_0 (fi      | 11 irst Rx/ ATA_2 ATA_4 | mode)  10                      | 9 ST    | Data  8 FDID[10 EXTID D_LAFN (TID_L/ | Bus 7 0:0] 0[15:0] 0[10:0] | Beferen  6   5  MSG MSG | _DAT/<br>_DAT/<br>_DAT/ | \_1<br>\_3<br>\_5 | 3       |       | EXTID | [17:16]      | Word  Access Size  Word/LW  Word  Word/LW  Word  Byte/Word/LW  Byte/Word/LW  Byte/Word/LW | Trigger Time Field Name Control 0 LAFM |

Figure 18.3 Mailbox-n Structure (continued)

- Notes: 1. All bits shadowed in grey are reserved and must be written LOW. The value returned by a read may not always be '0' and should not be relied upon.
  - 2. ATX and DART are not supported by Mailbox-0, and the MBC setting of Mailbox-0 is limited.
  - 3. ID Reorder (MCR15) can change the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

## (1) Message Control Field

**STDID**[10:0]: These bits set the identifier (standard identifier) of data frames and remote frames.

**EXTID**[17:0]: These bits set the identifier (extended identifier) of data frames and remote frames.

**RTR** (Remote Transmission Request bit): Used to distinguish between data frames and remote frames. This bit is overwritten by received CAN Frames depending on Data Frames or Remote Frames.

**Important:** Please note that, when ATX bit is set with the setting MBC = 001(bin), the RTR bit will never be set. When a Remote Frame is received, the CPU can be notified by the corresponding RFPR set or IRR[2] (Remote Frame Receive Interrupt), however, as RCAN-TL1 needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

**Important:** In order to support automatic answer to remote frame when MBC = 001 (bin) is used and ATX = 1 the RTR flag must be programmed to zero to allow data frame to be transmitted.

Note: when a Mailbox is configured to send a remote frame request the DLC used for transmission is the one stored into the Mailbox.

| RTR | Description  |
|-----|--------------|
| 0   | Data frame   |
| 1   | Remote frame |

**IDE** (Identifier Extension bit): Used to distinguish between the standard format and extended format of CAN data frames and remote frames.

| IDE | Description     |
|-----|-----------------|
| 0   | Standard format |
| 1   | Extended format |

### Mailbox-0

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

Note: MBC[1] of MB0 is always "1".

### Mailbox-31 to 1

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

NMC (New Message Control): When this bit is set to '0', the Mailbox of which the RXPR or RFPR bit is already set does not store the new message but maintains the old one and sets the UMSR correspondent bit. When this bit is set to '1', the Mailbox of which the RXPR or RFPR bit is already set overwrites with the new message and sets the UMSR correspondent bit.

**Important:** Please note that if a remote frame is overwritten with a data frame or vice versa could be that both RXPR and RFPR flags (together with UMSR) are set for the same Mailbox. In this case the RTR bit within the Mailbox Control Field should be relied upon.

**Important:** Please note that when the Time Triggered mode is used NMC needs to be set to '1' for Mailbox 31 to allow synchronization with all incoming reference messages even when RXPR[31] is not cleared.

| NMC | Description                  |  |  |  |  |
|-----|------------------------------|--|--|--|--|
| 0   | Overrun mode (Initial value) |  |  |  |  |
| 1   | Overwrite mode               |  |  |  |  |

**ATX** (Automatic Transmission of Data Frame): When this bit is set to '1' and a Remote Frame is received into the Mailbox DLC is stored. Then, a Data Frame is transmitted from the same Mailbox using the current contents of the message data and updated DLC by setting the corresponding TXPR automatically. The scheduling of transmission is still governed by ID priority or Mailbox priority as configured with the Message Transmission Priority control bit (MCR.2). In order to use this function, MBC[2:0] needs to be programmed to be '001' (Bin). When a transmission is performed by this function, the DLC (Data Length Code) to be used is the one that has been received. Application needs to guarantee that the DLC of the remote frame correspond to the DLC of the data frame requested.

**Important:** When ATX is used and MBC = 001 (Bin) the filter for the IDE bit cannot be used since ID of remote frame has to be exactly the same as that of data frame as the reply message.

**Important:** Please note that, when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the CPU will be notified by the corresponding RFPR set, however, as RCAN-TL1 needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

**Important:** Please note that in case of overrun condition (UMSR flag set when the Mailbox has its NMC = 0) the message received is discarded. In case a remote frame is causing overrun into a Mailbox configured with ATX = 1, the transmission of the corresponding data frame may be triggered only if the related PFPR flag is cleared by the CPU when the UMSR flag is set. In such case PFPR flag would get set again.

| ATX | Description   |
|-----|---|
| 0   | Automatic Transmission of Data Frame disabled (Initial value) |
| 1   | Automatic Transmission of Data Frame enabled                  |

**DART (Disable Automatic Re-Transmission):** When this bit is set, it disables the automatic retransmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is set to '0', RCAN-TL1 tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.

| DART | Description                             |
|------|---|
| 0    | Re-transmission enabled (Initial value) |
| 1    | Re-Transmission disabled                |

MBC[2:0] (Mailbox Configuration): These bits configure the nature of each Mailbox as follows. When MBC = 111 (Bin), the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. The MBC = '110', '101' and '100' settings are prohibited. When the MBC is set to any other value, the LAFM field becomes available. Please don't set TXPR when MBC is set as reception as there is no hardware protection, and TXPR will remain set. MBC[1] of Mailbox-0 is fixed to "1" by hardware. This is to ensure that MB0 cannot be configured to transmit Messages.

|        |        |        | Data<br>Frame | Remote<br>Frame | Data<br>Frame | Remote<br>Frame |   |
|--------|--------|--------|---------------|-----------------|---------------|-----------------|---|
| MBC[2] | MBC[1] | MBC[0] | Transmit      | Transmit        | Receive       | Receive         | Remarks                                       |
| 0      | 0      | 0      | Yes           | Yes             | No            | No              | Not allowed for Mailbox-0                     |
|        |        |        |               |                 |               |                 | <ul> <li>Time-Triggered</li> </ul>            |
|        |        |        |               |                 |               |                 | transmission can be used                      |
| 0      | 0      | 1      | Yes           | Yes             | No            | Yes             | Can be used with ATX*                         |
|        |        |        |               |                 |               |                 | <ul> <li>Not allowed for Mailbox-0</li> </ul> |
|        |        |        |               |                 |               |                 | <ul> <li>LAFM can be used</li> </ul>          |
| 0      | 1      | 0      | No            | No              | Yes           | Yes             | Allowed for Mailbox-0                         |
|        |        |        |               |                 |               |                 | <ul> <li>LAFM can be used</li> </ul>          |
| 0      | 1      | 1      | No            | No              | Yes           | No              | Allowed for Mailbox-0                         |
|        |        |        |               |                 |               |                 | <ul> <li>LAFM can be used</li> </ul>          |
| 1      | 0      | 0      |               |                 | Se            | tting prohib    | pited   |
| 1      | 0      | 1      |               |                 | Se            | tting prohib    | pited   |
| 1      | 1      | 0      |               |                 | Se            | tting prohib    | pited   |
| 1      | 1      | 1      |               |                 | Mailbox i     | nactive (In     | itial value)                                  |

Notes: \* In order to support automatic retransmission, RTR shall be "0" when MBC = 001(bin) and ATX = 1.

When ATX = 1 is used the filter for IDE must not be used.

**DLC**[3:0] (Data Length Code): These bits encode the number of data bytes from 0,1, 2, ... 8 that will be transmitted in a data frame. Please note that when a remote frame request is transmitted the DLC value to be used must be the same as the DLC of the data frame that is requested.

| DLC[3] | DLC[2] | DLC[1] | DLC[0] | Description                           |
|--------|--------|--------|--------|---------------------------------------|
| 0      | 0      | 0      | 0      | Data Length = 0 bytes (Initial value) |
| 0      | 0      | 0      | 1      | Data Length = 1 byte                  |
| 0      | 0      | 1      | 0      | Data Length = 2 bytes                 |
| 0      | 0      | 1      | 1      | Data Length = 3 bytes                 |
| 0      | 1      | 0      | 0      | Data Length = 4 bytes                 |
| 0      | 1      | 0      | 1      | Data Length = 5 bytes                 |
| 0      | 1      | 1      | 0      | Data Length = 6 bytes                 |
| 0      | 1      | 1      | 1      | Data Length = 7 bytes                 |
| 1      | х      | х      | х      | Data Length = 8 bytes                 |

#### **(2) Local Acceptance Filter Mask (LAFM)**

This area is used as Local Acceptance Filter Mask (LAFM) for receive boxes.

**LAFM:** When MBC is set to 001, 010, 011(Bin), this field is used as LAFM Field. It allows a Mailbox to accept more than one identifier. The LAFM is comprised of two 16-bit read/write areas as follows.

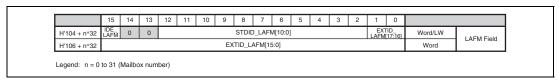


Figure 18.4 Acceptance filter

If a bit is set in the LAFM, then the corresponding bit of a received CAN identifier is ignored when the RCAN-TL1 searches a Mailbox with the matching CAN identifier. If the bit is cleared, then the corresponding bit of a received CAN identifier must match to the STDID/IDE/EXTID set in the mailbox to be stored. The structure of the LAFM is same as the message control in a Mailbox. If this function is not required, it must be filled with '0'.

**Important:** RCAN-TL1 starts to find a matching identifier from Mailbox-31 down to Mailbox-0. As soon as RCAN-TL1 finds one matching, it stops the search. The message will be stored or not depending on the NMC and RXPR/RFPR flags. This means that, even using LAFM, a received message can only be stored into 1 Mailbox.

**Important:** When a message is received and a matching Mailbox is found, the whole message is stored into the Mailbox. This means that, if the LAFM is used, the STDID, RTR, IDE and EXTID may differ to the ones originally set as they are updated with the STDID, RTR, IDE and EXTID of the received message.

**STD\_LAFM[10:0]** — Filter mask bits for the CAN base identifier [10:0] bits.

| STD_LAFM[10:0] | Description                               |
|----------------|---|
| 0              | Corresponding STD_ID bit is cared         |
| 1              | Corresponding STD_ID bit is "don't cared" |

**EXT LAFM[17:0]** — Filter mask bits for the CAN Extended identifier [17:0] bits.

| EXT_LAFM[17:0] | Description                               |
|----------------|---|
| 0              | Corresponding EXT_ID bit is cared         |
| 1              | Corresponding EXT_ID bit is "don't cared" |

**IDE LAFM** — Filter mask bit for the CAN IDE bit.

| IDE_LAFM | Description                            |
|----------|--|
| 0        | Corresponding IDE bit is cared         |
| 1        | Corresponding IDE bit is "don't cared" |

## (3) Message Data Fields

Page 832 of 1824

Storage for the CAN message data that is transmitted or received. MSG\_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 through to bit 0.

When CMAX!= 3'b111/MBC[30] = 3'b000 and TXPR[30] is set, Mailbox-30 is configured as transmission of time reference. Its DLC must be greater than 0 and its RTR must be zero (as specified for TTCAN Level 1) so that the Cycle\_count (CCR register) is embedded in the first byte of the data field instead of MSG\_DATA\_0[5:0] when this Mailbox starts transmission. This function shall be used when RCAN-TL1 is enabled to work in TTCAN mode to perform a Potential Time Master role to send the Time reference message. MSG\_DATA\_0[7:6] is still transmitted as stored in the Mailbox. User can set MSG\_DATA\_0[7] when a Next\_is\_Gap needs to be transmitted.

Please note that the CCR value is only embedded on the frame transmitted but not stored back into Mailbox 30.

When CMAX!= 3'b111, MBC[31] = 3'b011 and TXPR[31] is cleared, Mailbox-31 is configured as reception of time reference. When a valid reference message is received (DLC > 0) RCAN-TL1 performs internal synchronisation (modifying its RFMK and basic cycle CCR).

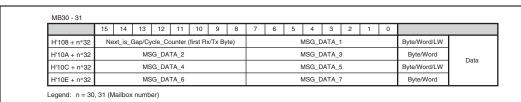
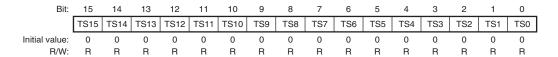


Figure 18.5 Message Data Field

#### **(4) Timestamp**

Storage for the Timestamp recorded on messages for transmit/receive. The Timestamp will be a useful function to monitor if messages are received/transmitted within expected schedule.

### **Timestamp**



Message Receive: For received messages of Mailbox-15 to 0, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle\_Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0) at SOF.

For messages received into Mailboxes 30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

Message Transmit: For transmitted messages of Mailbox-15 to 1, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0), at SOF.

For messages transmitted from Mailboxes 30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

**Important:** Please note that the TimeStamp is stored in a temporary register. Only after a successful transmission or reception the value is then copied into the related Mailbox field. The TimeStamp may also be updated if the CPU clears RXPR/RFPR at the same time that UMSR is set in overrun, however it can be read properly before clearing RXPR/RFPR.

### (5) Tx-Trigger Time (TTT) and Time Trigger control

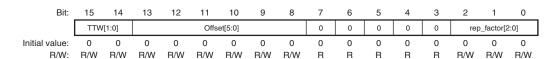
For Mailbox-29 to 24, when MBC is set to 000 (Bin) in time trigger mode (CMAX!= 3'b111), Tx-Trigger Time works as Time\_Mark to determine the boundary between time windows. The TTT and TT control are comprised of two 16-bit read/write areas as follows. Mailbox-30 doesn't have TT control and works as Time Ref.

Mailbox 30 to 24 can be used for reception if not used for transmission in TT mode. However they cannot join the event trigger transmission queue when the TT mode is used.

### Tx-Trigger Time

| Bit:           | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|----------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
|                | TTT15 | TTT14 | TTT13 | TTT12 | TTT11 | TTT10 | TTT9 | TTT8 | TTT7 | TTT6 | TTT5 | TTT4 | TTT3 | TTT2 | TTT1 | TTT0 |
| Initial value: | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| R/W:           | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |

# • Time Trigger control



The following figure shows the differences between all Mailboxes supporting Time Triggered mode.

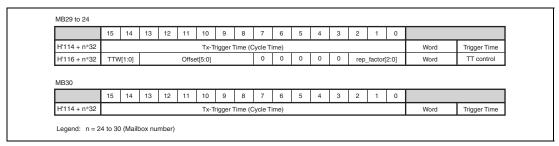


Figure 18.6 Tx-Trigger control field

TTW[1:0] (Time Trigger Window): These bits show the attribute of time windows. Please note that once a merged arbitrating window is opened by TTW = 2'b10, the window must be closed by TTW = 2'b11. Several messages with TTW = 2'b10 may be used within the start and the end of a merged arbitrating window.

| TTW[1] | TTW[0] | Description                        |
|--------|--------|------------------------------------|
| 0      | 0      | Exclusive window (initial value)   |
| 0      | 1      | Arbitrating window                 |
| 1      | 0      | Start of merged arbitrating window |
| 1      | 1      | End of merged arbitrating window   |

The first 16-bit area specifies the time that triggers the transmission of the message in cycle time. The second 16-bit area specifies the basic cycle in the system matrix where the transmission must start (Offset) and the frequency for periodic transmission. When the internal TTT register matches to the CYCTR value, and the internal Offset matches to CCR value transmission is attempted from the corresponding Mailbox. In order to enable this function, the CMAX (Cycle Maximum Register) must be set to a value different from 3'b111, the Timer (TCNTR) must be running (TTCR0 bit15 = 1), the corresponding MBC must be set to 3'b000 and the corresponding TXPR bit must be set. Once TXPR is set by S/W, RCAN-TL1 does not clear the corresponding TXPR bit (among Mailbox-30 to 24) to carry on performing the periodic transmission. In order to stop the periodic transmission, TXPR must be cleared by TXCR. Please note that in this case it is possible that both TXACK and ABACK are set for the same Mailbox if TXACK is not cleared right after completion of transmission. Please refer to Figure 18.7.

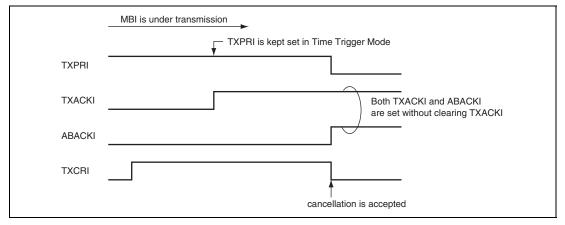


Figure 18.7 TXACK and ABACK in Time Trigger Transmission

Please note that for Mailbox 30 TTW is fixed to '01', Offset to '00' and rep\_factor to '0'. The following tables report the combinations for the rep\_factor and the offset.

| Rep_factor | Description  |
|------------|--|
| 3'b000     | Every basic cycle (initial value)                    |
| 3'b001     | Every two basic cycle                                |
| 3'b010     | Every four basic cycle                               |
| 3'b011     | Every eight basic cycle                              |
| 3'b100     | Every sixteen basic cycle                            |
| 3'b101     | Every thirty two basic cycle                         |
| 3'b110     | Every sixty four basic cycle (once in system matrix) |
| 3'b111     | Reserved   |

The Offset Field determines the first cycle in which a Time Triggered Mailbox may start transmitting its Message.

| Offset    | Description  |
|-----------|--|
| 6'b000000 | Initial Offset = 1 <sup>st</sup> Basic Cycle (initial value) |
| 6'b000001 | Initial Offset = 2 <sup>nd</sup> Basic Cycles                |
| 6'b000010 | Initial Offset = 3 <sup>rd</sup> Basic Cycles                |
| 6'b000011 | Initial Offset = 4 <sup>th</sup> Basic Cycles                |
| 6'b000100 | Initial Offset = 5 <sup>th</sup> Basic Cycles                |
|           |  |
|           |  |
| 6'b111110 | Initial Offset = 63 <sup>rd</sup> Basic Cycles               |
| 6'b111111 | Initial Offset = 64 <sup>th</sup> Basic Cycles               |

# The following relation must be maintained:

Cycle\_Count\_Maximum + 1 >= Repeat\_Factor > Offset

Cycle\_Count\_Maximum =  $2^{CMAX} - 1$ 

Repeat\_Factor =  $2^{\text{rep\_factor}}$ 

# CMAX, Repeat\_Factor, and Offset are register values

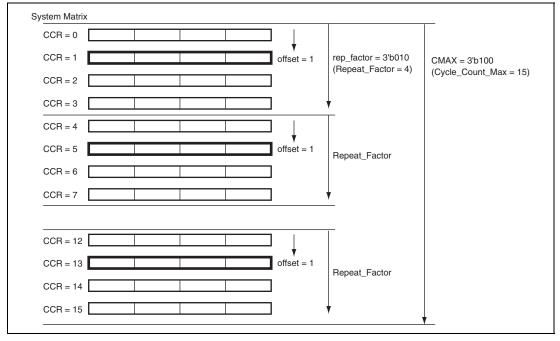


Figure 18.8 System Matrix

Tx-Trigger Times must be set in ascending order such that the difference between them satisfies the following condition.

TTT (mailbox i) - TTT (mailbox i - 1) > TEW + Maximum frame length + 9

# 18.3.3 RCAN-TL1 Control Registers

The following sections describe RCAN-TL1 control registers. The address is mapped as follow.

**Important:** These registers can only be accessed in Word size (16-bit).

| Description                              | Address | Name    | Access Size (bits) |
|--|---------|---------|--------------------|
| Master Control Register                  | 000     | MCR     | Word               |
| General Status Register                  | 002     | GSR     | Word               |
| Bit Configuration Register 1             | 004     | BCR1    | Word               |
| Bit Configuration Register 0             | 006     | BCR0    | Word               |
| Interrupt Register                       | 008     | IRR     | Word               |
| Interrupt Mask Register                  | 00A     | IMR     | Word               |
| Error Counter Register                   | 00C     | TEC/REC | Word               |
| Message Buffer Error Status Register     | 600     | MBESR   | Word               |
| Message Buffer Error Control<br>Register | 602     | MBECR   | Word               |

Figure 18.9 RCAN-TL1 control registers

# (1) Master Control Register (MCR)

The Master Control Register (MCR) is a 16-bit read/write register that controls RCAN-TL1.

• MCR (Address = H'000)

| Bit:           | 15    | 14    | 13 | 12 | 11 | 10  | 9        | 8   | 7    | 6    | 5    | 4 | 3 | 2    | 1    | 0    |
|----------------|-------|-------|----|----|----|-----|----------|-----|------|------|------|---|---|------|------|------|
|                | MCR15 | MCR14 | -  | -  | -  |     | TST[2:0] |     | MCR7 | MCR6 | MCR5 | - | - | MCR2 | MCR1 | MCR0 |
| Initial value: | 1     | 0     | 0  | 0  | 0  | 0   | 0        | 0   | 0    | 0    | 0    | 0 | 0 | 0    | 0    | 1    |
| R/W:           | R/W   | R/W   | R  | R  | R  | R/W | R/W      | R/W | R/W  | R/W  | R/W  | R | R | R/W  | R/W  | R/W  |

**Bit 15** — **ID Reorder** (MCR15): This bit changes the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

| Bit15: MCR15 | Description                                       |
|--------------|---|
| 0            | RCAN-TL1 is the same as HCAN2                     |
| 1            | RCAN-TL1 is not the same as HCAN2 (Initial value) |

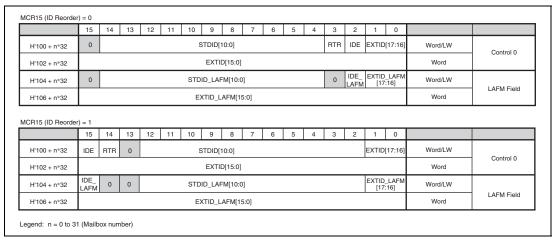


Figure 18.10 ID Reorder

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-TL1 enters BusOff.

| Bit14: MCR14 | Description   |
|--------------|---|
| 0            | RCAN-TL1 remains in BusOff for normal recovery sequence (128 x 11 Recessive Bits) (Initial value) |
| 1            | RCAN-TL1 moves directly into Halt Mode after it enters BusOff if MCR6 is set.                     |

This bit can be modified only in reset mode.

- Bit 13 Reserved. The written value should always be '0' and the returned value is '0'.
- Bit 12 Reserved. The written value should always be '0' and the returned value is '0'.
- Bit 11 Reserved. The written value should always be '0' and the returned value is '0'.
- **Bit 10 8 Test Mode** (**TST[2:0]**): This bit enables/disables the test modes. Please note that before activating the Test Mode it is requested to move RCAN-TL1 into Halt mode or Reset mode. This is to avoid that the transition to Test Mode could affect a transmission/reception in progress. For details, please refer to section 18.4.1, Test Mode Settings.

Please note that the test modes are allowed only for diagnosis and tests and not when RCAN-TL1 is used in normal operation.

| Bit10:<br>TST2 | Bit9:<br>TST1 | Bit8:<br>TST0 | Description                          |
|----------------|---------------|---------------|--------------------------------------|
| 0              | 0             | 0             | Normal Mode (initial value)          |
| 0              | 0             | 1             | Listen-Only Mode (Receive-Only Mode) |
| 0              | 1             | 0             | Self Test Mode 1 (External)          |
| 0              | 1             | 1             | Self Test Mode 2 (Internal)          |
| 1              | 0             | 0             | Write Error Counter                  |
| 1              | 0             | 1             | Error Passive Mode                   |
| 1              | 1             | 0             | Setting prohibited                   |
| 1              | 1             | 1             | Setting prohibited                   |

**Bit 7** — **Auto-wake Mode (MCR7):** MCR7 enables or disables the Auto-wake mode. If this bit is set, the RCAN-TL1 automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared the RCAN-TL1 does not automatically cancel the sleep mode.

RCAN-TL1 cannot store the message that wakes it up.

Note: This bit can be modified only Reset or Halt mode.

| Bit7: MCR7 | Description  |
|------------|--|
| 0          | Auto-wake by CAN bus activity disabled (Initial value) |
| 1          | Auto-wake by CAN bus activity enabled                  |

**Bit 6** — **Halt during Bus Off (MCR6):** MCR6 enables or disables entering Halt mode immediately when MCR1 is set during Bus Off. This bit can be modified only in Reset or Halt mode. Please note that when Halt is entered in Bus Off the CAN engine is also recovering immediately to Error Active mode.

| Bit6: MCR6 | Description  |
|------------|--|
| 0          | If MCR[1] is set, RCAN-TL1 will not enter Halt mode during Bus Off but wait up to end of recovery sequence (Initial value) |
| 1          | Enter Halt mode immediately during Bus Off if MCR[1] or MCR[14] are asserted.  |

**Bit 5** — **Sleep Mode (MCR5):** Enables or disables Sleep mode transition. If this bit is set, while RCAN-TL1 is in halt mode, the transition to sleep mode is enabled. Setting MCR5 is allowed after entering Halt mode. The two Error Counters (REC, TEC) will remain the same during Sleep mode. This mode will be exited in two ways:

- 1. by writing a '0' to this bit position,
- 2. or, if MCR[7] is enabled, after detecting a dominant bit on the CAN bus.

If Auto wake up mode is disabled, RCAN-TL1 will ignore all CAN bus activities until the sleep mode is terminated. When leaving this mode the RCAN-TL1 will synchronise to the CAN bus (by checking for 11 recessive bits) before joining CAN Bus activity. This means that, when the No.2 method is used, RCAN-TL1 will miss the first message to receive. CAN transceivers stand-by mode will also be unable to cope with the first message when exiting stand by mode, and the S/W needs to be designed in this manner.

In sleep mode only the following registers can be accessed: MCR, GSR, IRR and IMR.

**Important:** RCAN-TL1 is required to be in Halt mode before requesting to enter in Sleep mode. That allows the CPU to clear all pending interrupts before entering sleep mode. Once all interrupts are cleared RCAN-TL1 must leave the Halt mode and enter Sleep mode simultaneously (by writing MCR[5] = 1 and MCR[1] = 0 at the same time).

| Bit 5: MCR5 | Description                                  |
|-------------|--|
| 0           | RCAN-TL1 sleep mode released (Initial value) |
| 1           | Transition to RCAN-TL1 sleep mode enabled    |

- Bit 4 Reserved. The written value should always be '0' and the returned value is '0'.
- Bit 3 Reserved. The written value should always be '0' and the returned value is '0'.
- **Bit 2 Message Transmission Priority** (MCR2): MCR2 selects the order of transmission for pending transmit data. If this bit is set, pending transmit data are sent in order of the bit position in the Transmission Pending Register (TXPR). The order of transmission starts from Mailbox-31 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission). Please note that this feature cannot be used for time trigger transmission of the Mailboxes 24 to 30.

If MCR2 is cleared, all messages for transmission are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field (STDID + IDE bit + EXTID (if IDE = 1) + RTR bit) with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit (internal arbitration works in the same

way as the arbitration on the CAN Bus between two CAN nodes starting transmission at the same time).

This bit can be modified only in Reset or Halt mode.

| Bit 2: MCR2 | Description   |
|-------------|---|
| 0           | Transmission order determined by message identifier priority (Initial value)                  |
| 1           | Transmission order determined by mailbox number priority (Mailbox-31 $\rightarrow$ Mailbox-1) |

**Bit 1—Halt Request (MCR1):** Setting the MCR1 bit causes the CAN controller to complete its current operation and then enter Halt mode (where it is cut off from the CAN bus). The RCAN-TL1 remains in Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interface does not join the CAN bus activity and does not store messages or transmit messages. All the user registers (including Mailbox contents and TEC/REC) remain unchanged with the exception of IRR0 and GSR4 which are used to notify the halt status itself. If the CAN bus is in idle or intermission state regardless of MCR6, RCAN-TL1 will enter Halt Mode within one Bit Time. If MCR6 is set, a halt request during Bus Off will be also processed within one Bit Time. Otherwise the full Bus Off recovery sequence will be performed beforehand. Entering the Halt Mode can be notified by IRR0 and GSR4.

If both MCR14 and MCR6 are set, MCR1 is automatically set as soon as RCAN-TL1 enters BusOff.

In the Halt mode, the RCAN-TL1 configuration can be modified with the exception of the Bit Timing setting, as it does not join the bus activity. MCR[1] has to be cleared by writing a '0' in order to re-join the CAN bus. After this bit has been cleared, RCAN-TL1 waits until it detects 11 recessive bits, and then joins the CAN bus.

- Notes: 1. After issuing a Halt request the CPU is not allowed to set TXPR or TXCR or clear MCR1 until the transition to Halt mode is completed (notified by IRR0 and GSR4). After MCR1 is set this can be cleared only after entering Halt mode or through a reset operation (SW or HW).
  - 2. Transition into or recovery from HALT mode, is only possible if the BCR1 and BCR0 registers are configured to a proper Baud Rate.

| Bit 1: MCR1 | Description                        |  |  |  |  |  |  |  |
|-------------|------------------------------------|--|--|--|--|--|--|--|
| 0           | Clear Halt request (Initial value) |  |  |  |  |  |  |  |
| 1           | Halt mode transition request       |  |  |  |  |  |  |  |

**Bit 0** — **Reset Request (MCR0):** Controls resetting of the RCAN-TL1 module. When this bit is changed from '0' to '1' the RCAN-TL1 controller enters its reset routine, re-initializing the internal logic, which then sets GSR3 and IRR0 to notify the reset mode. During a re-initialization, all user registers are initialized.

RCAN-TL1 can be re-configured while this bit is set. This bit has to be cleared by writing a '0' to join the CAN bus. After this bit is cleared, the RCAN-TL1 module waits until it detects 11 recessive bits, and then joins the CAN bus. The Baud Rate needs to be set up to a proper value in order to sample the value on the CAN Bus.

After Power On Reset, this bit and GSR3 are always set. This means that a reset request has been made and RCAN-TL1 needs to be configured.

The Reset Request is equivalent to a Power On Reset but controlled by Software.

| Bit 0: MCR0 | Description   |
|-------------|---|
| 0           | Clear Reset Request   |
| 1           | CAN Interface reset mode transition request (Initial value) |

# (2) General Status Register (GSR)

The General Status Register (GSR) is a 16-bit read-only register that indicates the status of RCAN-TL1.

• GSR (Address = H'002)

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5    | 4    | 3    | 2    | 1    | 0    |
|----------------|----|----|----|----|----|----|---|---|---|---|------|------|------|------|------|------|
|                | -  | -  | -  | -  | -  | -  | - | - | - | - | GSR5 | GSR4 | GSR3 | GSR2 | GSR1 | GSR0 |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0    | 0    | 1    | 1    | 0    | 0    |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R | R | R | R    | R    | R    | R    | R    | R    |

Bits 15 to 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 5 — Error Passive Status Bit (GSR5): Indicates whether the CAN Interface is in Error Passive or not. This bit will be set high as soon as the RCAN-TL1 enters the Error Passive state and is cleared when the module enters again the Error Active state (this means the GSR5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR5 and GSR0 must be considered.

| Bit 5: GSR5 | Description   |
|-------------|---|
| 0           | RCAN-TL1 is not in Error Passive or in Bus Off status (Initial value)                               |
|             | [Reset condition] RCAN-TL1 is in Error Active state   |
| 1           | RCAN-TL1 is in Error Passive (if GSR0 = 0) or Bus Off (if GSR0 = 1)                                 |
|             | [Setting condition] When TEC $\geq$ 128 or REC $\geq$ 128 or if Error Passive Test Mode is selected |

Bit 4 — Halt/Sleep Status Bit (GSR4): Indicates whether the CAN engine is in the halt/sleep state or not. Please note that the clearing time of this flag is not the same as the setting time of IRR12.

Please note that this flag reflects the status of the CAN engine and not of the full RCAN-TL1 IP. RCAN-TL1 exits sleep mode and can be accessed once MCR5 is cleared. The CAN engine exits sleep mode only after two additional transmission clocks on the CAN Bus.

| Bit 4: GSR4 | Description  |
|-------------|--|
| 0           | RCAN-TL1 is not in the Halt state or Sleep state (Initial value)   |
| 1           | Halt mode (if MCR1 = 1) or Sleep mode (if MCR5 = 1)  |
|             | [Setting condition] If MCR1 is set and the CAN bus is either in intermission or idle or MCR5 is set and RCAN-TL1 is in the halt mode or RCAN-TL1 is moving to Bus Off when MCR14 and MCR6 are both set |

Bit 3 — Reset Status Bit (GSR3): Indicates whether the RCAN-TL1 is in the reset state or not.

| Bit 3: GSR3 | Description  |
|-------------|--|
| 0           | RCAN-TL1 is not in the reset state   |
| 1           | Reset state (Initial value)  |
|             | [Setting condition] After an RCAN-TL1 internal reset (due to SW or HW reset) |

**Bit 2** — **Message Transmission in progress Flag (GSR2):** Flag that indicates to the CPU if the RCAN-TL1 is in Bus Off or transmitting a message or an error/overload flag due to error detected during transmission. The timing to set TXACK is different from the time to clear GSR2. TXACK is set at the 7<sup>th</sup> bit of End Of Frame. GSR2 is set at the 3<sup>rd</sup> bit of intermission if there are no more messages ready to be transmitted. It is also set by arbitration lost, bus idle, reception, reset or halt transition.

| Bit 2: GSR2 | Description  |
|-------------|--|
| 0           | RCAN-TL1 is in Bus Off or a transmission is in progress                            |
| 1           | [Setting condition] Not in Bus Off and no transmission in progress (Initial value) |

Bit 1—Transmit/Receive Warning Flag (GSR1): Flag that indicates an error warning.

| Bit 1: GSR1 | Description   |
|-------------|---|
| 0           | [Reset condition] When (TEC < 96 and REC < 96) or Bus Off (Initial value) |
| 1           | [Setting condition] When $96 \le TEC < 256$ or $96 \le REC < 256$         |

Note: REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence. However the flag GSR1 is not set in Bus Off.

Bit 0—Bus Off Flag (GSR0): Flag that indicates that RCAN-TL1 is in the bus off state.

| Bit 0: GSR0 | Description   |
|-------------|---|
| 0           | [Reset condition]   |
|             | Recovery from bus off state or after a HW or SW reset (Initial value) |
| 1           | [Setting condition] When TEC ≥ 256 (bus off state)                    |

Note: Only the lower 8 bits of TEC are accessible from the user interface. The 9<sup>th</sup> bit is equivalent to GSR0.

# (3) Bit Configuration Register (BCR0, BCR1)

The bit configuration registers (BCR0 and BCR1) are 2 X 16-bit read/write register that are used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN Interface.

The Time quanta is defined as:

$$Timequanta = \frac{2 * BRP}{f_{clk}}$$

Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral bus frequency.

#### • BCR1 (Address = H'004)

| Bit:           | 15  | 14  | 13     | 12  | 11 | 10  | 9        | 8   | 7 | 6 | 5   | 4      | 3 | 2 | 1 | 0   |
|----------------|-----|-----|--------|-----|----|-----|----------|-----|---|---|-----|--------|---|---|---|-----|
|                |     | TSG | 1[3:0] |     | -  | -   | TSG2[2:0 | )]  | - | - | SJW | /[1:0] | - | - | - | BSP |
| Initial value: | 0   | 0   | 0      | 0   | 0  | 0   | 0        | 0   | 0 | 0 | 0   | 0      | 0 | 0 | 0 | 0   |
| R/W:           | R/W | R/W | R/W    | R/W | R  | R/W | R/W      | R/W | R | R | R/W | R/W    | R | R | R | R/W |

**Bits 15 to 12** — **Time Segment 1 (TSG1[3:0] = BCR1[15:12]):** These bits are used to set the segment TSEG1 (= PRSEG + PHSEG1) to compensate for edges on the CAN Bus with a positive phase error. A value from 4 to 16 time quanta can be set.

Bit 15: Bit 14: Bit 13: Bit 12: TSG1[3] TSG1[2] TSG1[1] TSG1[0] Description

| 0 | 0 | 0 | 0 | Setting prohibited (Initial value) |
|---|---|---|---|------------------------------------|
| 0 | 0 | 0 | 1 | Setting prohibited                 |
| 0 | 0 | 1 | 0 | Setting prohibited                 |
| 0 | 0 | 1 | 1 | PRSEG + PHSEG1 = 4 time quanta     |
| 0 | 1 | 0 | 0 | PRSEG + PHSEG1 = 5 time quanta     |
| : | : | : | : | :                                  |
| : | : | : | : | :                                  |
| 1 | 1 | 1 | 1 | PRSEG + PHSEG1 = 16 time quanta    |
| _ |   |   |   |                                    |

Bit 11: Reserved. The written value should always be '0' and the returned value is '0'.

**Bits 10 to 8** — **Time Segment 2 (TSG2[2:0] = BCR1[10:8]):** These bits are used to set the segment TSEG2 (= PHSEG2) to compensate for edges on the CAN Bus with a negative phase error. A value from 2 to 8 time quanta can be set as shown below.

**Bit 8:** 

Bit 9:

Bit 10:

| TSG2[2] | TSG2[1] | TSG2[0] | Description                                       |
|---------|---------|---------|---|
| 0       | 0       | 0       | Setting prohibited (Initial value)                |
| 0       | 0       | 1       | PHSEG2 = 2 time quanta (conditionally prohibited) |
| 0       | 1       | 0       | PHSEG2 = 3 time quanta                            |
| 0       | 1       | 1       | PHSEG2 = 4 time quanta                            |
| 1       | 0       | 0       | PHSEG2 = 5 time quanta                            |
| 1       | 0       | 1       | PHSEG2 = 6 time quanta                            |
| 1       | 1       | 0       | PHSEG2 = 7 time quanta                            |
| 1       | 1       | 1       | PHSEG2 = 8 time quanta                            |

Bits 7 and 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bits 5 and 4 - ReSynchronisation Jump Width (SJW[1:0] = BCR0[5:4]): These bits set the synchronisation jump width.

| Bit 5:<br>SJW[1] | Bit 4:<br>SJW[0] | Description   |
|------------------|------------------|---|
| 0                | 0                | Synchronisation Jump width = 1 time quantum (Initial value) |
| 0                | 1                | Synchronisation Jump width = 2 time quanta                  |
| 1                | 0                | Synchronisation Jump width = 3 time quanta                  |
| 1                | 1                | Synchronisation Jump width = 4 time quanta                  |

Bits 3 to 1: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 — Bit Sample Point (BSP = BCR1[0]): Sets the point at which data is sampled.

| Bit 0 : BSP | Description   |
|-------------|---|
| 0           | Bit sampling at one point (end of time segment 1) (Initial value)                   |
| 1           | Bit sampling at three points (rising edge of the last three clock cycles of PHSEG1) |

• BCR0 (Address = H'006)

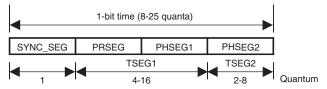
| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6   | 5   | 4   | 3     | 2   | 1   | 0   |
|----------------|----|----|----|----|----|----|---|---|-----|-----|-----|-----|-------|-----|-----|-----|
| [              | -  | -  | -  | -  | -  | -  | - | - |     |     |     | BRP | [7:0]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

Bits 8 to 15: Reserved. The written value should always be '0' and the returned value is '0'.

Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]): These bits are used to define the peripheral bus clock periods contained in a Time Quantum.

| Bit 7:<br>BRP[7] | Bit 6:<br>BRP[6] | Bit 5:<br>BRP[5] | Bit 4:<br>BRP[4] | Bit 3:<br>BRP[3] | Bit 2:<br>BRP[2] | Bit 1:<br>BRP[1] | Bit 0:<br>BRP[0] | Description                                   |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|---|
| 0                | 0                | 0                | 0                | 0                | 0                | 0                | 0                | 2 X peripheral bus clock<br>(Initial value)   |
| 0                | 0                | 0                | 0                | 0                | 0                | 0                | 1                | 4 X peripheral bus clock                      |
| 0                | 0                | 0                | 0                | 0                | 0                | 1                | 0                | 6 X peripheral bus clock                      |
| :<br>:           | :                | :                | :                | :                | :                | :                | :                | 2*(register value + 1) X peripheral bus clock |
| 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 512 X peripheral bus clock                    |

# • Requirements of Bit Configuration Register



SYNC\_SEG: Segment for establishing synchronisation of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks.

PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended

when synchronisation (resynchronisation) is established.)

PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened

when synchronisation (resynchronisation) is established)

TSEG1: TSG1 + 1

TSEG2: TSG2 + 1

The RCAN-TL1 Bit Rate Calculation is:

Bit Rate = 
$$\frac{f_{Clk}}{2 \times (BRP + 1) \times (TSEG1 + TSEG2 + 1)}$$

Where BRP is given by the register value and TSEG1 and TSEG2 are derived values from TSG1 and TSG2 register values. The '+1' in the above formula is for the Sync-Seg which duration is 1 time quanta.

 $f_{CLK}$  = Peripheral Clock

**BCR Setting Constraints** 

$$TSEG1min > TSEG2 \ge SJWmax$$
 (SJW = 1 to 4)

 $8 \le TSEG1 + TSEG2 + 1 \le 25$  time quanta (TSEG1 + TSEG2 + 1 = 7 is not allowed)

 $TSEG2 \ge 2$ 

These constraints allow the setting range shown in the table below for TSEG1 and TSEG2 in the Bit Configuration Register. The number in the table shows possible setting of SJW. "No" shows that there is no allowed combination of TSEG1 and TSEG2.

|      |       | 001 | 010 | 011 | 100 | 101 | 110 | 111 | TSG2  |
|------|-------|-----|-----|-----|-----|-----|-----|-----|-------|
|      |       | 2   | 3   | 4   | 5   | 6   | 7   | 8   | TSEG2 |
| TSG1 | TSEG1 |     |     |     |     |     |     |     |       |
| 0011 | 4     | No  | 1-3 | No  | No  | No  | No  | No  |       |
| 0100 | 5     | 1-2 | 1-3 | 1-4 | No  | No  | No  | No  |       |
| 0101 | 6     | 1-2 | 1-3 | 1-4 | 1-4 | No  | No  | No  |       |
| 0110 | 7     | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | No  | No  |       |
| 0111 | 8     | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | No  |       |
| 1000 | 9     | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |       |
| 1001 | 10    | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |       |
| 1010 | 11    | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |       |
| 1011 | 12    | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |       |
| 1100 | 13    | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |       |
| 1101 | 14    | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |       |
| 1110 | 15    | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |       |
| 1111 | 16    | 1-2 | 1-3 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |       |

**Example 1:** To have a Bit rate of 500Kbps with a frequency of fclk = 40MHz it is possible to set: BRP = 3, TSEG1 = 6, TSEG2 = 3.

Then the configuration to write is BCR1 = 5200 and BCR0 = 0003.

**Example 2:** To have a Bit rate of 250Kps with a frequency of 35MHz it is possible to set: BPR = 4, TSEG1 = 8, TSEG2 = 5.

Then the configuration to write is BCR1 = 7400 and BCR0 = 0004.

# (4) Interrupt Request Register (IRR)

The interrupt register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

• IRR (Address = H'008)

| Bit:           | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|----------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
|                | IRR15 | IRR14 | IRR13 | IRR12 | IRR11 | IRR10 | IRR9 | IRR8 | IRR7 | IRR6 | IRR5 | IRR4 | IRR3 | IRR2 | IRR1 | IRR0 |
| Initial value: | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1    |
| R/W:           | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R    | R    | R/W  | R/W  | R/W  | R/W  | R/W  | R    | R    | R/W  |

Bit 15 — Timer Compare Match Interrupt 1 (IRR15): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time (TCMR1 = CYCTR), this bit is set.

| Bit 15: IRR15 | Description   |
|---------------|---|
| 0             | Timer Compare Match has not occurred to the TCMR1 (Initial value) |
|               | [Clearing condition] Writing 1                                    |
| 1             | Timer Compare Match has occurred to the TCMR1                     |
|               | [Setting condition] TCMR1 matches to Cycle Time (TCMR1 = CYCTR)   |

Bit 14 — Timer Compare Match Interrupt 0 (IRR14): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to Local Time (TCMR0 = TCNTR), this bit is set.

| Bit 14: IRR14 | Description  |
|---------------|--|
| 0             | Timer Compare Match has not occurred to the TCMR0 (Initial value)    |
|               | [Clearing condition] Writing 1                                       |
| 1             | Timer Compare Match has occurred to the TCMR0                        |
|               | [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR) |

**Bit 13 - Timer Overrun Interrupt/Next\_is\_Gap Reception Interrupt/Message Error Interrupt (IRR13):** This interrupt assumes a different meaning depending on the RCAN-TL1 mode. It indicates that:

— The Timer (TCNTR) has overrun when RCAN-TL1 is working in event-trigger mode (including test modes)

- Time reference message with Next\_is\_Gap set has been received when working in time-trigger mode. Please note that when a Next\_is\_Gap is received the application is responsible to stop all transmission at the end of the current basic cycle (including test modes)
- Message error has occurred when in test mode. Note: If a Message Overload condition occurs when in Test Mode, then this bit will not be set.

| Bit 13: IRR13 | Description   |
|---------------|---|
| 0             | Timer (TCNTR) has not overrun in event-trigger mode (including test modes) (Initial value)                |
|               | Time reference message with Next_is_Gap has not been received in time-trigger mode (including test modes) |
|               | Message error has not occurred in test mode   |
|               | [Clearing condition] Writing 1  |
| 1             | [Setting condition]   |
|               | Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-trigger mode (including test modes)  |
|               | Time reference message with Next_is_Gap has been received in time-trigger mode (including test modes)     |
|               | Message error has occurred in test mode   |

**Bit 12 – Bus activity while in sleep mode (IRR12):** IRR12 indicates that a CAN bus activity is present. While the RCAN-TL1 is in sleep mode and a dominant bit is detected on the CAN bus, this bit is set. This interrupt is cleared by writing a '1' to this bit position. Writing a '0' has no effect. If auto wakeup is not used and this interrupt is not requested it needs to be disabled by the related interrupt mask register. If auto wake up is not used and this interrupt is requested it should be cleared only after recovering from sleep mode. This is to avoid that a new falling edge of the reception line causes the interrupt to get set again.

Please note that the setting time of this interrupt is different from the clearing time of GSR4.

| Bit 12: IRR12 | Description   |
|---------------|---|
| 0             | Bus idle state (Initial value)                                  |
|               | [Clearing condition] Writing 1                                  |
| 1             | CAN bus activity detected in RCAN-TL1 sleep mode                |
|               | [Setting condition]   |
|               | Dominant bit level detection on the Rx line while in sleep mode |

Bit 11 — Timer Compare Match Interrupt 2 (IRR11): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time (TCMR2 = CYCTR), this bit is set.

| Bit 11: IRR11 | Description   |
|---------------|---|
| 0             | Timer Compare Match has not occurred to the TCMR2 (initial value) |
|               | [Clearing condition] Writing 1                                    |
| 1             | Timer Compare Match has occurred to the TCMR2                     |
|               | [Setting condition] TCMR2 matches to Cycle Time (TCMR2 = CYCTR)   |

Bit 10 — Start of new system matrix Interrupt (IRR10): Indicates that a new system matrix is starting.

When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message. Please note that when CMAX = 0 this interrupt is set at every basic cycle.

| Bit 10: IRR10 | Description  |  |  |  |  |  |  |  |
|---------------|--|--|--|--|--|--|--|--|
| 0             | A new system matrix is not starting (initial value)  |  |  |  |  |  |  |  |
|               | [Clearing condition] Writing 1   |  |  |  |  |  |  |  |
| 1             | Cycle counter reached zero.  |  |  |  |  |  |  |  |
|               | [Setting condition]  |  |  |  |  |  |  |  |
|               | Reception/transmission of time reference message is successfully completed when CMAX!= $3$ 'b111 and CCR = $0$ |  |  |  |  |  |  |  |

Bit 9 – Message Overrun/Overwrite Interrupt Flag (IRR9): Flag indicating that a message has been received but the existing message in the matching Mailbox has not been read as the corresponding RXPR or RFPR is already set to '1' and not yet cleared by the CPU. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (New Message Control) bit. This bit is cleared when all bit in UMSR (Unread Message Status Register) are cleared (by writing '1') or by setting MBIMR (MailBox interrupt Mast Register) for all UMSR flag set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

| Bit 9: IRR9 | Description   |  |  |  |  |  |  |  |
|-------------|---|--|--|--|--|--|--|--|
| 0           | No pending notification of message overrun/overwrite  |  |  |  |  |  |  |  |
|             | [Clearing condition]  |  |  |  |  |  |  |  |
|             | Clearing of all bit in UMSR/setting MBIMR for all UMSR set (initial value)                      |  |  |  |  |  |  |  |
| 1           | A receive message has been discarded due to overrun condition or a message has been overwritten |  |  |  |  |  |  |  |
|             | [Setting condition]   |  |  |  |  |  |  |  |
|             | Message is received while the corresponding RXPR and/or RFPR = 1 and MBIMR = $0$                |  |  |  |  |  |  |  |

**Bit 8 - Mailbox Empty Interrupt Flag (IRR8):** This bit is set when one of the messages set for transmission has been successfully sent (corresponding TXACK flag is set) or has been successfully aborted (corresponding ABACK flag is set). In Event Triggered mode the related TXPR is also cleared and this mailbox is now ready to accept a new message data for the next transmission. In Time Trigger mode TXPR for the Mailboxes from 30 to 24 is not cleared after a successful transmission in order to keep transmitting at each programmed basic cycle. In effect, this bit is set by an OR'ed signal of the TXACK and ABACK bits not masked by the corresponding MBIMR flag. Therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

| Bit 8: IRR8 | Description  |  |  |  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|--|--|--|
| 0           | Messages set for transmission or transmission cancellation request NOT progressed. (Initial value)   |  |  |  |  |  |  |  |  |
|             | [Clearing Condition]   |  |  |  |  |  |  |  |  |
|             | All the TXACK and ABACK bits are cleared/setting MBIMR for all TXACK and ABACK set   |  |  |  |  |  |  |  |  |
| 1           | Message has been transmitted or aborted, and new message can be stored (in TT mode Mailbox 24 to 30 can be programmed with a new message only in case of abortion) |  |  |  |  |  |  |  |  |
|             | [Setting condition]  |  |  |  |  |  |  |  |  |
|             | When a TXACK or ABACK bit is set (if related MBIMR = 0).   |  |  |  |  |  |  |  |  |

**Bit 7 - Overload Frame (IRR7):** Flag indicating that the RCAN-TL1 has detected a condition that should initiate the transmission of an overload frame. Note that in the condition of transmission being prevented, such as listen only mode, an Overload Frame will NOT be transmitted, but IRR7 will still be set. IRR7 remains asserted until reset by writing a '1' to this bit position - writing a '0' has no effect.

| Bit 7: IRR7 | Description                                      |
|-------------|--|
| 0           | [Clearing condition] Writing 1 (Initial value)   |
| 1           | [Setting conditions] Overload condition detected |

**Bit 6 - Bus Off Interrupt Flag (IRR6):** This bit is set when RCAN-TL1 enters the Bus-off state or when RCAN-TL1 leaves Bus-off and returns to Error-Active. The cause therefore is the existing condition TEC ≥ 256 at the node or the end of the Bus-off recovery sequence (128X11 consecutive recessive bits) or the transition from Bus Off to Halt (automatic or manual). This bit remains set even if the RCAN-TL1 node leaves the bus-off condition, and needs to be explicitly cleared by S/W. The S/W is expected to read the GSR0 to judge whether RCAN-TL1 is in the bus-off or error active status. It is cleared by writing a '1' to this bit position even if the node is still bus-off. Writing a '0' has no effect.

| Bit 6: IRR6 | Description  |  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|--|
| 0           | [Clearing condition] Writing 1 (Initial value)   |  |  |  |  |  |  |
| 1           | Enter Bus off state caused by transmit error or Error Active state returning from Bus-off                                |  |  |  |  |  |  |
|             | [Setting condition]  |  |  |  |  |  |  |
|             | When TEC becomes $\geq$ 256 or End of Bus-off after 128X11 consecutive recessive bits or transition from Bus Off to Halt |  |  |  |  |  |  |

**Bit 5 - Error Passive Interrupt Flag (IRR5):** Interrupt flag indicating the error passive state caused by the transmit or receive error counter or by Error Passive forced by test mode. This bit is reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the node may still be error passive. Please note that the SW needs to check GSR0 and GSR5 to judge whether RCAN-TL1 is in Error Passive or Bus Off status.

| Bit 5: IRR5 | Description  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|
| 0           | [Clearing condition] Writing 1 (Initial value)                           |  |  |  |  |  |
| 1           | Error passive state caused by transmit/receive error                     |  |  |  |  |  |
|             | [Setting condition]  |  |  |  |  |  |
|             | When TEC $\geq$ 128 or REC $\geq$ 128 or Error Passive test mode is used |  |  |  |  |  |
|             |  |  |  |  |  |  |

**Bit 4 - Receive Error Counter Warning Interrupt Flag (IRR4):** This bit becomes set if the receive error counter (REC) reaches a value greater than 95 when RCAN-TL1 is not in the Bus Off status. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

| Bit 4: IRR4 | Description  |
|-------------|--|
| 0           | [Clearing condition] Writing 1 (Initial value)                   |
| 1           | Error warning state caused by receive error                      |
|             | [Setting condition] When REC ≥ 96 and RCAN-TL1 is not in Bus Off |

Bit 3 - Transmit Error Counter Warning Interrupt Flag (IRR3): This bit becomes set if the transmit error counter (TEC) reaches a value greater than 95. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

| Bit 3: IRR3 | Description                                    |
|-------------|--|
| 0           | [Clearing condition] Writing 1 (Initial value) |
| 1           | Error warning state caused by transmit error   |
|             | [Setting condition] When TEC ≥ 96              |

**Bit 2 - Remote Frame Receive Interrupt Flag (IRR2):** Flag indicating that a remote frame has been received in a mailbox. This bit is set if at least one receive mailbox, with related MBIMR not set, contains a remote frame transmission request. This bit is automatically cleared when all bits in the Remote Frame Receive Pending Register (RFPR), are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

| Bit 2: IRR2 | Description   |  |  |  |  |  |  |  |
|-------------|---|--|--|--|--|--|--|--|
| 0           | [Clearing condition] Clearing of all bits in RFPR (Initial value) |  |  |  |  |  |  |  |
| 1           | At least one remote request is pending                            |  |  |  |  |  |  |  |
|             | [Setting condition]   |  |  |  |  |  |  |  |
|             | When remote frame is received and the corresponding MBIMR = 0     |  |  |  |  |  |  |  |

**Bit 1 – Data Frame Received Interrupt Flag (IRR1):** IRR1 indicates that there are pending Data Frames received. If this bit is set at least one receive mailbox contains a pending message. This bit is cleared when all bits in the Data Frame Receive Pending Register (RXPR) are cleared, i.e. there is no pending message in any receiving mailbox. It is in effect a logical OR of the RXPR flags from each configured receive mailbox with related MBIMR not set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

| Bit 1: IRR1 | Description   |
|-------------|---|
| 0           | [Clearing condition] Clearing of all bits in RXPR (Initial value)         |
| 1           | Data frame received and stored in Mailbox                                 |
|             | [Setting condition] When data is received and the corresponding MBIMR = 0 |

Bit 0 – Reset/Halt/Sleep Interrupt Flag (IRR0): This flag can get set for three different reasons. It can indicate that:

- 1. Reset mode has been entered after a SW (MCR0) or HW reset
- 2. Halt mode has been entered after a Halt request (MCR1)
- 3. Sleep mode has been entered after a sleep request (MCR5) has been made while in Halt mode.

The GSR may be read after this bit is set to determine which state RCAN-TL1 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 18.15 Halt Mode/Sleep Mode.

IRR0 is set by the transition from "0" to "1" of GSR3 or GSR4 or by transition from Halt mode to Sleep mode. So, IRR0 is not set if RCAN-TL1 enters Halt mode again right after exiting from Halt mode, without GSR4 being cleared. Similarly, IRR0 is not set by direct transition from Sleep mode to Halt Request. At the transition from Halt/Sleep mode to Transition/Reception, clearing GSR4 needs (one-bit time - TSEG2) to (one-bit time \* 2 - TSEG2).

In the case of Reset mode, IRR0 is set, however, the interrupt to the CPU is not asserted since IMR0 is automatically set by initialization.

| Bit 0: IRR0 | Description  |  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|--|
| 0           | [Clearing condition] Writing 1   |  |  |  |  |  |  |
| 1           | Transition to S/W reset mode or transition to halt mode or transition to sleep mode (Initial value)                            |  |  |  |  |  |  |
|             | [Setting condition]  |  |  |  |  |  |  |
|             | When reset/halt/sleep transition is completed after a reset (MCR0 or HW) or Halt mode (MCR1) or Sleep mode (MCR5) is requested |  |  |  |  |  |  |

#### Message Buffer Error Status Register (MBESR) (5)

This register is a status register that indicates the generation of a parity error in RAM (control 0, LAFM, and DATA). When a parity error occurs, the MBEF bit is set to 1.

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

Bit 15 to 1: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 – Message Buffer Parity Error Detection Status Flag (MBEF): Indicates the occurrence of a parity error with a data transfer. This bit is cleared by writing 1. Writing 0 to this bit is ignored.

The MBESR register is not initialized when the value of the MCR0 bit in the master Note: control register changes from 0 to 1.

| Bit 0: MBEF | Description                 |
|-------------|-----------------------------|
| 0           | Parity error does not occur |
| 1           | Parity error occurs         |

# (6) Interrupt Mask Register (IMR)

The interrupt mask register is a 16 bit register that protects all corresponding interrupts in the Interrupt Request Register (IRR) from generating an output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to '1'. This register can be read or written at any time. The IMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the IRR.

# • IMR (Address = H'00A)

| Bit:           | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|----------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
|                | IMR15 | IMR14 | IMR13 | IMR12 | IMR11 | IMR10 | IMR9 | IMR8 | IMR7 | IMR6 | IMR5 | IMR4 | IMR3 | IMR2 | IMR1 | IMR0 |
| Initial value: | 1     | 1     | 1     | 1     | 1     | 1     | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    |
| R/W:           | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |

**Bit 15 to 0:** Maskable interrupt sources corresponding to IRR[15:0] respectively. When a bit is set, the interrupt signal is not generated, although setting the corresponding IRR bit is still performed.

| Bit[15:0]: IMRn | Description   |
|-----------------|---|
| 0               | Corresponding IRR is not masked (IRQ is generated for interrupt conditions) |
| 1               | Corresponding interrupt of IRR is masked (Initial value)                    |

# (7) Message Buffer Error Control Register (MBECR)

This register is a readable/writable register that protects an interrupt in message buffer error status register (MBESR). An interrupt request is masked when the MBIM bit is set to 1. The MBIM bit controls the interrupt request, but does not prevent the setting of the MBIM bit.



Bit 15 to 1: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 – Message Buffer Parity Error Detection Interrupt Mask (MBIM): When a bit is set, the interrupt signal is not generated, although setting the MBEF bit is still performed.

Note: The MBECR register is not initialized when the value of the MCR0 bit in the master control register changes from 0 to 1.

| Bit 0: MBIM | Description                  |
|-------------|------------------------------|
| 0           | MBEF interrupt is not masked |
| 1           | MBEF interrupt is masked     |

### (8) Transmit Error Counter (TEC) and Receive Error Counter (REC)

The Transmit Error Counter (TEC) and Receive Error Counter (REC) is a 16-bit read/(write) register that functions as a counter indicating the number of transmit/receive message errors on the CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [1], [2], [3] and [4]. When not in (Write Error Counter) test mode this register is read only, and can only be modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) or entering to bus off.

In Write Error Counter test mode (i.e. TST[2:0] = 3'b100), it is possible to write to this register. The same value can only be written to TEC/REC, and the value written into TEC is set to TEC and REC. When writing to this register, RCAN-TL1 needs to be put into Halt Mode. This feature is only intended for test purposes.

• TEC/REC (Address = H'00C)

| Bit:           | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    | 7    | 6    | 5      | 4    | 3    | 2    | 1    | 0    |
|----------------|------|------|------|------|------|------|------|------|------|------|--------|------|------|------|------|------|
|                | TEC7 | TEC6 | TEC5 | TEC4 | TEC3 | TEC2 | TEC1 | TEC0 | REC7 | REC6 | REC5   | REC4 | REC3 | REC2 | REC1 | REC0 |
| Initial value: | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0      | 0    | 0    | 0    | 0    | 0    |
| B/W-           | R/M* | R/M∗ | R/M∗ | R/M∗ | R/M∗ | R/W* | R/M* | R/M* | R/M* | R/M* | R/\//∗ | R/M* | R/M* | R/M* | R/M∗ | R/M∗ |

Note: \* It is only possible to write the value in test mode when TST[2:0] in MCR is 3'b100.

REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence.

# 18.3.4 RCAN-TL1 Mailbox Registers

The following sections describe RCAN-TL1 Mailbox registers that control/flag individual Mailboxes. The address is mapped as follows.

Important: LongWord access is carried out as two consecutive Word accesses.

# 32-Mailboxes version

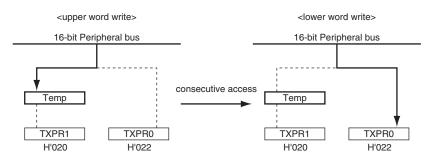
| Description                       | Address | Name   | Access Size (bits) |
|-----------------------------------|---------|--------|--------------------|
| Transmit Pending 1                | 020     | TXPR1  | LW                 |
| Transmit Pending 0                | 022     | TXPR0  | _                  |
|                                   | 024     |        |                    |
|                                   | 026     |        |                    |
| Transmit Cancel 1                 | 028     | TXCR1  | Word/LW            |
| Transmit Cancel 0                 | 02A     | TXCR0  | Word               |
|                                   | 02C     |        |                    |
|                                   | 02E     |        |                    |
| Transmit Acknowledge 1            | 030     | TXACK1 | Word/LW            |
| Transmit Acknowledge 0            | 032     | TXACK0 | Word               |
|                                   | 034     |        |                    |
|                                   | 036     |        |                    |
| Abort Acknowledge 1               | 038     | ABACK1 | Word/LW            |
| Abort Acknowledge 0               | 03A     | ABACK0 | Word               |
|                                   | 03C     |        |                    |
|                                   | 03E     |        |                    |
| Data Frame Receive Pending 1      | 040     | RXPR1  | Word/LW            |
| Data Frame Receive Pending 0      | 042     | RXPR0  | Word               |
|                                   | 044     |        |                    |
|                                   | 046     |        |                    |
| Remote Frame Receive Pending 1    | 048     | RFPR1  | Word/LW            |
| Remote Frame Receive Pending 0    | 04A     | RFPR0  | Word               |
|                                   | 04C     |        |                    |
|                                   | 04E     |        |                    |
| Mailbox Interrupt Mask Register 1 | 050     | MBIMR1 | Word/LW            |
| Mailbox Interrupt Mask Register 0 | 052     | MBIMR0 | Word               |
|                                   | 054     |        |                    |
|                                   | 056     |        |                    |
| Unread message Status Register 1  | 058     | UMSR1  | Word/LW            |
| Unread message Status Register 0  | 05A     | UMSR0  | Word               |
|                                   | 05C     |        |                    |
|                                   | 05E     |        |                    |

Figure 18.11 RCAN-TL1 Mailbox registers

### (1) Transmit Pending Register (TXPR1, TXPR0)

The concatenation of TXPR1 and TXPR0 is a 32-bit register that contains any transmit pending flags for the CAN module. In the case of 16-bit bus interface, Long Word access is carried out as two consecutive word accesses.

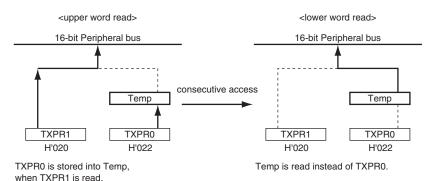
### <Longword Write Operation>



Data is stored into Temp instead of TXPR1.

Longword data are stored into both TXPR1 and TXPR0 at the same time.

#### <Longword Read Operation>



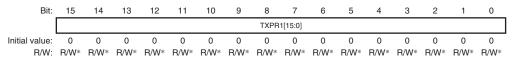
The TXPR1 controls Mailbox-31 to Mailbox-16, and the TXPR0 controls Mailbox-15 to Mailbox-1. The CPU may set the TXPR bits to affect any message being considered for transmission by writing a '1' to the corresponding bit location. Writing a '0' has no effect, and TXPR cannot be cleared by writing a '0' and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the CPU to determine which, if any, transmissions are pending or in progress. In effect there is a transmit pending bit for all Mailboxes except for the Mailbox-0. Writing a '1' to a bit location when the mailbox is not configured to transmit is not allowed.

In Event Triggered Mode RCAN-TL1 will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. In Time Trigger Mode, TXPR for the Mailboxes from 30 to 24 is NOT cleared after a successful transmission, in order to keep transmitting at each programmed basic cycle. The TXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and RCAN-TL1 automatically tries to transmit it again unless its DART bit (Disable Automatic Re-Transmission) is set in the Message-Control of the corresponding Mailbox. In such case (DART set), the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR8) and the correspondent bit within the Abort Acknowledgement Register (ABACK).

If the status of the TXPR changes, the RCAN-TL1 shall ensure that in the identifier priority scheme (MCR2 = 0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to the Application Note for details.

When the RCAN-TL1 changes the state of any TXPR bit position to a '0', an empty slot interrupt (IRR8) may be generated. This indicates that either a successful or an aborted mailbox transmission has just been made. If a message transmission is successful it is signalled in the TXACK register, and if a message transmission abortion is successful it is signalled in the ABACK register. By checking these registers, the contents of the Message of the corresponding Mailbox may be modified to prepare for the next transmission.

#### TXPR1

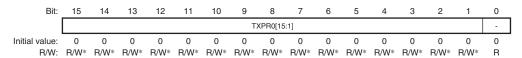


Note: \* It is possible only to write a '1' for a Mailbox configured as transmitter.

**Bit 15 to 0** — Requests the corresponding Mailbox to transmit a CAN Frame. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

| Bit[15:0]: TXPR1 | Description  |
|------------------|--|
| 0                | Transmit message idle state in corresponding mailbox (Initial value)   |
|                  | [Clearing Condition]   |
|                  | Completion of message transmission (for Event Triggered Messages) or message transmission abortion (automatically cleared) |
| 1                | Transmission request made for corresponding mailbox  |

#### TXPR0



Note: \* It is possible only to write a '1' for a Mailbox configured as transmitter.

**Bit 15 to 1** — Indicates that the corresponding Mailbox is requested to transmit a CAN Frame. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

| Bit[15:1]: TXPR0 | Description  |
|------------------|--|
| 0                | Transmit message idle state in corresponding mailbox (Initial value)   |
|                  | [Clearing Condition]   |
|                  | Completion of message transmission (for Event Triggered Messages) or message transmission abortion (automatically cleared) |
| 1                | Transmission request made for corresponding mailbox  |

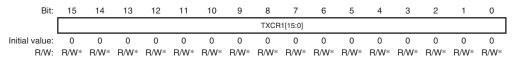
**Bit 0— Reserved:** This bit is always '0' as this is a receive-only Mailbox. Writing a '1' to this bit position has no effect. The returned value is '0'.

#### **(2)** Transmit Cancel Register (TXCR1, TXCR0)

The TXCR1 and TXCR0 are 16-bit read/conditionally-write registers. The TXCR1 controls Mailbox-31 to Mailbox-16, and the TXCR0 controls Mailbox-15 to Mailbox-1. This register is used by the CPU to request the pending transmission requests in the TXPR to be cancelled. To clear the corresponding bit in the TXPR the CPU must write a '1' to the bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXCR bits, and sets the corresponding ABACK bit. However, once a Mailbox has started a transmission, it cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABACK bit. If an attempt is made by the CPU to clear a mailbox transmission that is not transmit-pending it has no effect. In this case the CPU will be not able at all to set the TXCR flag.

#### TXCR1



Only writing a '1' to a Mailbox that is requested for transmission and is configured as Note: transmit.

Bit 15 to 0 — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 0 corresponds to Mailbox-31 to 16 (and TXPR1[15:0]) respectively.

| Bit[15:0]:TXCR1 | Description   |
|-----------------|---|
| 0               | Transmit message cancellation idle state in corresponding mailbox (Initial value) |
|                 | [Clearing Condition]  |
|                 | Completion of transmit message cancellation (automatically cleared)               |
| 1               | Transmission cancellation request made for corresponding mailbox                  |

#### TXCR0

| Bit:           | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8       | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0 |
|----------------|------|------|------|------|------|------|------|---------|------|------|------|------|------|------|------|---|
|                |      |      |      |      |      |      | T    | KCR0[15 | :1]  |      |      |      |      |      |      | - |
| Initial value: | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0       | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0 |
| R/W:           | R/W*    | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R |

Note: \* Only writing a '1' to a Mailbox that is requested for transmission and is configured as transmit.

**Bit 15 to 1** — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 1 corresponds to Mailbox-15 to 1 (and TXPR0[15:1]) respectively.

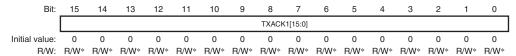
| Bit[15:1]: TXCR0 | Description   |
|------------------|---|
| 0                | Transmit message cancellation idle state in corresponding mailbox (Initial value) |
|                  | [Clearing Condition]  |
|                  | Completion of transmit message cancellation (automatically cleared)               |
| 1                | Transmission cancellation request made for corresponding mailbox                  |

**Bit 0** — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

# (3) Transmit Acknowledge Register (TXACK1, TXACK0)

The TXACK1 and TXACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been successfully made. When a transmission has succeeded the RCAN-TL1 sets the corresponding bit in the TXACK register. The CPU may clear a TXACK bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect.

#### TXACK1

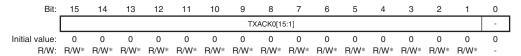


Note: \* Only when writing a '1' to clear.

**Bit 15 to 0** — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

| 0 | [Clearing Condition] Writing '1' (Initial value)                                  |
|---|---|
| 1 | Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) |
|   | [Setting Condition]   |
|   | Completion of message transmission for corresponding mailbox                      |

#### • TXACK0



Note: \* Only when writing a '1' to clear.

**Bit 15 to 1** — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

| Bit[15:1]   | 1:TXACK0    | Description |
|-------------|-------------|-------------|
| Dit[ 13. I] | J. I AACINO | Description |

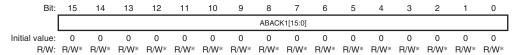
| 0 | [Clearing Condition] Writing '1' (Initial value)                                  |
|---|---|
| 1 | Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) |
|   | [Setting Condition]   |
|   | Completion of message transmission for corresponding mailbox                      |

**Bit 0** — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

## (4) Abort Acknowledge Register (ABACK1, ABACK0)

The ABACK1 and ABACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded the RCAN-TL1 sets the corresponding bit in the ABACK register. The CPU may clear the Abort Acknowledge bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect. An ABACK bit position is set by the RCAN-TL1 to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

#### ABACK1



Note: \* Only when writing a '1' to clear.

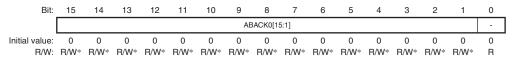
**Bit 15 to 0** — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

## Bit[15:0]:ABACK1 Description

| 0 | [Clearing Condition] Writing '1' (Initial value)                                   |
|---|--|
| 1 | Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) |
|   | [Setting Condition]  |
|   | Completion of transmission cancellation for corresponding mailbox                  |

#### ABACK0

Page 870 of 1824



Note: \* Only when writing a '1' to clear.

**Bit 15 to 1** — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

D:4[4E.4]. AD ACKO

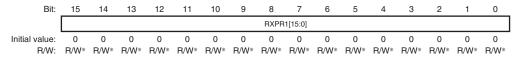
| BIT[15:1]:ABACKU | Description  |
|------------------|--|
| 0                | [Clearing Condition] Writing '1' (Initial value)                                   |
| 1                | Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) |
|                  | [Setting Condition]  |
|                  | Completion of transmission cancellation for corresponding mailbox                  |

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

#### Data Frame Receive Pending Register (RXPR1, RXPR0) (5)

The RXPR1 and RXPR0 are 16-bit read/conditionally-write registers. The RXPR is a register that contains the received Data Frames pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is set in the RXPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data Frame Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by receiving Data Frames and not by receiving Remote frames.

#### RXPR1

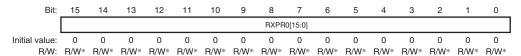


Only when writing a '1' to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 31 to 16 respectively.

| Bit[15:0]: RXPR1 | Description   |
|------------------|---|
| 0                | [Clearing Condition] Writing '1' (Initial value)                              |
| 1                | Corresponding Mailbox received a CAN Data Frame                               |
|                  | [Setting Condition] Completion of Data Frame receive on corresponding mailbox |

## • RXPR0



Note: \* Only when writing a '1' to clear.

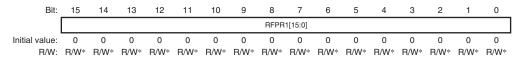
**Bit 15 to 0** — Configurable receive mailbox locations corresponding to each mailbox position from 15 to 0 respectively.

| Bit[15:0]: RXPR0 | Description   |
|------------------|---|
| 0                | [Clearing Condition] Writing '1' (Initial value)                              |
| 1                | Corresponding Mailbox received a CAN Data Frame                               |
|                  | [Setting Condition] Completion of Data Frame receive on corresponding mailbox |

#### (6) Remote Frame Receive Pending Register (RFPR1, RFPR0)

The RFPR1 and RFPR0 are 16-bit read/conditionally-write registers. The RFPR is a register that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is set in the RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. In effect there is a bit position for all mailboxes. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a RFPR bit is set, it also sets IRR2 (Remote Frame Receive Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Please note that these bits are only set by receiving Remote Frames and not by receiving Data frames.

#### RFPR1

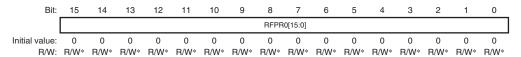


Note: \* Only when writing a '1' to clear.

**Bit 15 to 0** — Remote Request pending flags for mailboxes 31 to 16 respectively.

| Bit[15:0]: RFPR1 | Description   |
|------------------|---|
| 0                | [Clearing Condition] Writing '1' (Initial value)                                |
| 1                | Corresponding Mailbox received Remote Frame                                     |
|                  | [Setting Condition] Completion of remote frame receive in corresponding mailbox |

#### RFPR0



Note: \* Only when writing a '1' to clear.

**Bit 15 to 0** — Remote Request pending flags for mailboxes 15 to 0 respectively.

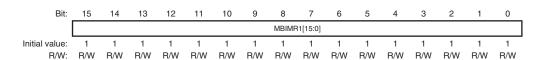
| Bit[15:0]: RFPR0 | Description   |
|------------------|---|
| 0                | [Clearing Condition] Writing '1' (Initial value)                                |
| 1                | Corresponding Mailbox received Remote Frame                                     |
|                  | [Setting Condition]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

## (7) Mailbox Interrupt Mask Register (MBIMR)

The MBIMR1 and MBIMR0 are 16-bit read/write registers. The MBIMR only prevents the setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received Interrupt, IRR[2] – Remote Frame Receive Interrupt, IRR[8] – Mailbox Empty Interrupt, and IRR[9] – Message OverRun/OverWrite Interrupt. If a mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR[2] and IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or UMSR. Similarly when a mailbox has been configured for transmission, a mask prevents the generation of an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or abortion of transmission (IRR[8]), however, it does not prevent the RCAN-TL1 from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and it does not prevent the RCAN-TL1 from clearing the corresponding TXPR/TXCR bit + setting the ABACK bit for abortion of the transmission.

A mask is set by writing a '1' to the corresponding bit position for the mailbox activity to be masked. At reset all mailbox interrupts are masked.

#### MBIMR1

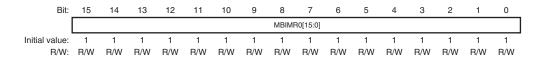


Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-31 to Mailbox-16 respectively.

# Bit[15:0]: MBIMR1 Description

| 0 | Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled                  |
|---|---|
| 1 | Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value) |

#### MBIMR0



Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-15 to Mailbox-0 respectively.

## Bit[15:0]: MBIMR0 Description

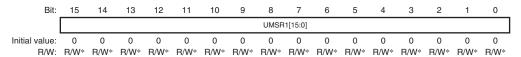
| 0 | Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled                  |
|---|---|
| 1 | Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value) |

#### (8) Unread Message Status Register (UMSR)

This register is a 32-bit read/conditionally write register and it records the mailboxes whose contents have not been accessed by the CPU prior to a new message being received. If the CPU has not cleared the corresponding bit in the RXPR or RFPR when a new message for that mailbox is received, the corresponding UMSR bit is set to '1'. This bit may be cleared by writing a '1' to the corresponding bit location in the UMSR. Writing a '0' has no effect.

If a mailbox is configured as transmit box, the corresponding UMSR will not be set.

#### UMSR1

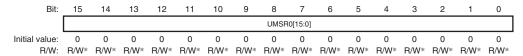


Note: \* Only when writing a '1' to clear.

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 31 to 16.

| Bit[15:0]: UMSR1 | Description  |
|------------------|--|
| 0                | [Clearing Condition] Writing '1' (initial value)                             |
| 1                | Unread received message is overwritten by a new message or overrun condition |
|                  | [Setting Condition]  |
|                  | When a new message is received before RXPR or RFPR is cleared                |

#### UMSR0



Note: \* Only when writing a '1' to clear.

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 15 to 0.

| Bit[15:0]: UMSR0 | Description  |
|------------------|--|
| 0                | [Clearing Condition] Writing '1' (initial value)                             |
| 1                | Unread received message is overwritten by a new message or overrun condition |
|                  | [Setting Condition]  |
|                  | When a new message is received before RXPR or RFPR is cleared                |

# 18.3.5 Timer Registers

The Timer is 16 bits and supports several source clocks. A pre-scale counter can be used to reduce the speed of the clock. It also supports three Compare Match Registers (TCMR2, TCMR1, TCMR0). The address map is as follows.

**Important:** These registers can only be accessed in Word size (16-bit).

| Description                                | Address | Name     | Access Size (bits) |
|--|---------|----------|--------------------|
| TimerTrigger Control Register 0            | 080     | TTCR0    | Word (16)          |
| Cycle Maximum/Tx-Enable<br>Window Register | 084     | CMAX_TEW | Word (16)          |
| Reference Trigger Offset Register          | 086     | RFTROFF  | Word (16)          |
| Timer Status Register                      | 088     | TSR      | Word (16)          |
| Cycle Counter Register                     | 08A     | CCR      | Word (16)          |
| Timer Counter Register                     | 08C     | TCNTR    | Word (16)          |
| Cycle Time Register                        | 090     | CYCTR    | Word (16)          |
| Reference Mark Register                    | 094     | RFMK     | Word (16)          |
| Timer Compare Match Register 0             | 098     | TCMR0    | Word (16)          |
| Timer Compare Match Register 1             | 09C     | TCMR1    | Word (16)          |
| Timer Compare Match Register 2             | 0A0     | TCMR2    | Word (16)          |
| Tx-Trigger Time Selection Register         | 0A4     | TTTSEL   | Word (16)          |

Figure 18.12 RCAN-TL1 Timer registers

## (1) Time Trigger Control Register (TTCR0)

The Time Trigger Control Register0 is a 16-bit read/write register and provides functions to control the operation of the Timer. When operating in Time Trigger Mode, please refer to section 18.4.3 (1), Time Triggered Transmission.

• TTCR0 (Address = H'080)

| Bit:           | 15    | 14    | 13    | 12    | 11    | 10    | 9 | 8 | 7 | 6    | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|-------|-------|-------|-------|-------|-------|---|---|---|------|-------|-------|-------|-------|-------|-------|
|                | TCR15 | TCR14 | TCR13 | TCR12 | TCR11 | TCR10 | - | - | - | TCR6 | TPSC5 | TPSC4 | TPSC3 | TPSC2 | TPSC1 | TPSC0 |
| Initial value: | 0     | 0     | 0     | 0     | 0     | 0     | 0 | 0 | 0 | 0    | 0     | 0     | 0     | 0     | 0     | 0     |
| B/W-           | R/W   | R/M   | R/M   | R/M   | R/M   | R/M   | R | R | R | R/M  | R/M   | R/W   | R/W   | R/W   | R/M   | R/M   |

**Bit 15** — **Enable Timer:** When this bit is set, the timer TCNTR is running. When this bit is cleared. TCNTR and CCR are cleared.

| Bit15: TTCR0 15 | Description  |
|-----------------|--|
| 0               | Timer and CCR are cleared and disabled (initial value) |
| 1               | Timer is running                                       |

**Bit 14** — **TimeStamp value:** Specifies if the Timestamp for transmission and reception in Mailboxes 15 to 1 must contain the Cycle Time (CYCTR) or the concatenation of CCR[5:0] + CYCTR[15:6]. This feature is very useful for time triggered transmission to monitor Rx\_Trigger.

This register does not affect the TimeStamp for Mailboxes 30 and 31.

| Bit14: TTCR0 14 | Description  |
|-----------------|--|
| 0               | CYCTR[15:0] is used for the TimeStamp in Mailboxes 15 to 1 (initial value) |
| 1               | CCR[5:0] + CYCTR[15:6] is used for the TimeStamp in Mailboxes 15 to 1      |

Bit 13 — Cancellation by TCMR2: The messages in the transmission queue are cancelled by setting TXCR, when both this bit and bit12 are set and compare match occurs when RCAN-TL1 is not in the Halt status, causing the setting of all TXCR bits with the corresponding TXPR bits set.

| Bit13: TTCR0 13 | Description   |
|-----------------|---|
| 0               | Cancellation by TCMR2 compare match is disabled (initial value) |
| 1               | Cancellation by TCMR2 compare match is enabled                  |

**Bit 12** — TCMR2 compare match enable: When this bit is set, IRR11 is set by TCMR2 compare match.

| Bit12 TTCR0 12 | Description  |
|----------------|--|
| 0              | IRR11 isn't set by TCMR2 compare match (initial value) |
| 1              | IRR11 is set by TCMR2 compare match                    |

Bit 11 — TCMR1 compare match enable: When this bit is set, IRR15 is set by TCMR1 compare match.

| Bit11 TTCR0 11 | Description  |
|----------------|--|
| 0              | IRR15 isn't set by TCMR1 compare match (initial value) |
| 1              | IRR15 is set by TCMR1 compare match                    |

Bit 10 — TCMR0 compare match enable: When this bit is set, IRR14 is set by TCMR0 compare match.

| Bit10 TTCR0 10 | Description  |
|----------------|--|
| 0              | IRR14 isn't set by TCMR0 compare match (initial value) |
| 1              | IRR14 is set by TCMR0 compare match                    |

Bits 9 to 7: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 6 — Timer Clear-Set Control by TCMR0: Specifies if the Timer is to be cleared and set to H'0000 when the TCMR0 matches to the TCNTR. Please note that the TCMR0 is also capable to generate an interrupt signal to the CPU via IRR14.

Note: If RCAN-TL1 is working in TTCAN mode (CMAX isn't 3'b111), TTCR0 bit6 has to be '0' to avoid clearing Local Time.

| Bit6: TTCR0 6 | Description                                       |
|---------------|---|
| 0             | Timer is not cleared by the TCMR0 (initial value) |
| 1             | Timer is cleared by the TCMR0                     |

**Bit5 to 0** — **RCAN-TL1 Timer Prescaler (TPSC[5:0]):** This control field allows the timer source clock (4\*[RCAN-TL1 system clock]) to be divided before it is used for the timer. This function is available only in event-trigger mode. In time trigger mode (CMAX is not 3'b111), one nominal Bit Timing (= one bit length of CAN bus) is automatically chosen as source clock of TCNTR.

The following relationship exists between source clock period and the timer period.

| Bit[5:0]: TPSC[5:0] | Description                      |
|---------------------|----------------------------------|
| 00000               | 1 X Source Clock (initial value) |
| 000001              | 2 X Source Clock                 |
| 000010              | 3 X Source Clock                 |
| 000011              | 4 X Source Clock                 |
| 000100              | 5 X Source Clock                 |
|                     |                                  |
|                     | •••••                            |
| 111111              | 64 X Source Clock                |

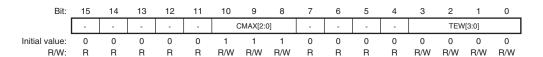
## (2) Cycle Maximum/Tx-Enable Window Register (CMAX\_TEW)

This register is a 16-bit read/write register. CMAX specifies the maximum value for the cycle counter (CCR) for TT Transmissions to set the number of basic cycles in the matrix system. When the Cycle Counter reaches the maximum value (CCR = CMAX), after a full basic cycle, it is cleared to zero and an interrupt is generated on IRR.10.

TEW specifies the width of Tx-Enable window.

• CMAX TEW (Address = H'084)

Page 880 of 1824



Bits 15 to 11: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 10 to 8 — Cycle Count Maximum (CMAX): Indicates the maximum number of CCR. The number of basic cycles available in the matrix cycle for Timer Triggered transmission is (Cycle Count Maximum + 1).

Unless CMAX = 3'b111, RCAN-TL1 is in time-trigger mode and time trigger function is available. If CMAX = 3'b111, RCAN-TL1 is in event-trigger mode.

| Bit[10:8]: CMAX[2:0] | Description   |
|----------------------|---|
| 000                  | Cycle Count Maximum = 0   |
| 0 0 1                | Cycle Count Maximum = 1   |
| 0 1 0                | Cycle Count Maximum = 3   |
| 0 1 1                | Cycle Count Maximum = 7   |
| 100                  | Cycle Count Maximum = 15  |
| 1 0 1                | Cycle Count Maximum = 31  |
| 110                  | Cycle Count Maximum = 63  |
| 111                  | CCR is cleared and RCAN-TL1 is in event-trigger mode. (initial value) |

**Important:** Please set CMAX = 3'b111 when event-trigger mode is used.

Bits 7 to 4: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 3 to 0 — Tx-Enable Window (TEW): Indicates the width of Tx-Enable Window. TEW = H'00 shows the width is one nominal Bit Timing. All values from 0 to 15 are allowed to be set.

| Bit[3:0]: TEW[3:0] | Description                                       |
|--------------------|---|
| 0000               | The width of Tx-Enable Window = 1 (initial value) |
| 0 0 0 1            | The width of Tx-Enable Window = 2                 |
| 0010               | The width of Tx-Enable Window = 3                 |
| 0 0 1 1            | The width of Tx-Enable Window = 4                 |
|                    |   |
|                    |   |
| 1111               | The width of Tx-Enable Window = 16                |

The CAN core always needs a time between 1 to 2 bit timing to initiate transmission. The above values are not considering this accuracy.

## (3) Reference Trigger Offset Register (RFTROFF)

This is a 8-bit read/write register that affects Tx-Trigger Time (TTT) of Mailbox-30. The TTT of Mailbox-30 is compared with CYCTR after RFTROFF extended with sign is added to the TTT. However, the value of TTT is not modified. The offset value doesn't affect others except Mailbox-30.

## • RFTROFF (Address = H'086)

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

Bit 15 to 8 — Indicate the value of Reference Trigger Offset.

Bits 7 to 0: Reserved. The written value should always be '0' and the returned value is '0'.

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | Description                             |
|-------|-------|-------|-------|-------|-------|------|------|---|
| 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | Ref_trigger_offset = +0 (initial value) |
| 0     | 0     | 0     | 0     | 0     | 0     | 0    | 1    | Ref_trigger_offset = +1                 |
| 0     | 0     | 0     | 0     | 0     | 0     | 1    | 0    | Ref_trigger_offset = +2                 |
|       | -     |       |       |       |       |      |      |   |
| 0     | 1     | 1     | 1     | 1     | 1     | 1    | 1    | Ref_trigger_offset = +127               |
|       | •     |       |       | •     |       |      | •    |   |
| 1     | 1     | 1     | 1     | 1     | 1     | 1    | 1    | Ref_trigger_offset = -1                 |
| 1     | 1     | 1     | 1     | 1     | 1     | 1    | 0    | Ref_trigger_offset = -2                 |
| •     |       |       |       |       |       |      |      |   |
| 1     | 0     | 0     | 0     | 0     | 0     | 0    | 1    | Ref_trigger_offset = -127               |
| 1     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | Prohibited                              |

#### **(4) Timer Status Register (TSR)**

This register is a 16-bit read-only register, and allows the CPU to monitor the Timer Compare Match status and the Timer Overrun Status.

TSR (Address = H'088)

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4    | 3    | 2    | 1    | 0    |
|----------------|----|----|----|----|----|----|---|---|---|---|---|------|------|------|------|------|
|                | -  | -  | -  | -  | -  | -  | - | - | - | - | - | TSR4 | TSR3 | TSR2 | TSR1 | TSR0 |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0    | 0    | 0    | 0    | 0    |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R    | R    | R    | R    | R    |

Bits 15 to 5: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 4 to 0 — RCAN-TL1 Timer Status (TSR[4:0]): This read-only field allows the CPU to monitor the status of the Cycle Counter, the Timer and the Compare Match registers. Writing to this field has no effect.

Bit 4 — Start of New System Matrix (TSR4): Indicates that a new system matrix is starting. When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message.

| Bit4: TSR4 | Description   |  |  |  |  |  |  |  |  |
|------------|---|--|--|--|--|--|--|--|--|
| 0          | A new system matrix is not starting (initial value)   |  |  |  |  |  |  |  |  |
|            | [Clearing condition] Writing '1' to IRR10 (Cycle Counter Overflow Interrupt)  |  |  |  |  |  |  |  |  |
| 1          | Cycle counter reached zero [Setting condition] When the Cycle Counter value changes from the maximum value (CMAX) to H'0. Reception/transmission of time reference message is successfully completed when CMAX!= 3'b111 and CCR = 0 |  |  |  |  |  |  |  |  |

Bit 3 — Timer Compare Match Flag 2 (TSR3): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time Register (TCMR2 = CYCTR), this bit is set if TTCR0 bit12 = 1. Please note that this bit is read-only and is cleared when IRR11 (Timer Compare Match Interrupt 2) is cleared.

| Bit3: TSR3 | Description   |
|------------|---|
| 0          | Timer Compare Match has not occurred to the TCMR2 (Initial value) [Clearing condition] Writing '1' to IRR11 (Timer Compare Match Interrupt 1) |
| 1          | Timer Compare Match has occurred to the TCMR2 [Setting condition]   |
|            | TCMR2 matches to Cycle Time (TCMR2 = CYCTR), if TTCR0 bit12 = 1.  |

Bit 2 — Timer Compare Match Flag 1 (TSR2): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time Register (TCMR1 = CYCTR), this bit is set if TTCR0 bit11 = 1. Please note that this bit is read-only and is cleared when IRR15 (Timer Compare Match Interrupt 1) is cleared.

| Bit2: TSR2 | Description   |
|------------|---|
| 0          | Timer Compare Match has not occurred to the TCMR1 (Initial value) [Clearing condition] Writing '1' to IRR15 (Timer Compare Match Interrupt 1) |
| 1          | Timer Compare Match has occurred to the TCMR1 [Setting condition]   |
|            | TCMR1 matches to Cycle Time (TCMR1 = CYCTR), if TTCR0 bit11 = 1.  |

**Bit 1** — **Timer Compare Match Flag 0 (TSR1):** Indicates that a Compare-Match condition occurred to the Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to the Timer value (TCMR0 = TCNTR), this bit is set if TTCR0 bit10 = 1. Please note that this bit is read-only and is cleared when IRR14 (Timer Compare Match Interrupt 0) is cleared.

| Bit1: TSR1 | Description   |
|------------|---|
| 0          | Compare Match has not occurred to the TCMR0 (Initial value) [Clearing condition] Writing '1' to IRR14 (Timer Compare Match Interrupt 0) |
| 1          | Compare Match has occurred to the TCMR0 [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR)                            |

Bit 0 — Timer Overrun/Next\_is\_Gap Reception/Message Error (TSR0): This flag is assigned to three different functions. It indicates that the Timer has overrun when working in event-trigger mode, time reference message with Next\_is\_Gap set has been received in time-trigger mode, and error detected on the CAN bus has occurred in test mode, respectively. Test mode has higher priority with respect to the other settings.

| Bit0: TSR0 | Description   |
|------------|---|
| 0          | Timer (TCNTR) has not overrun in event-trigger mode (Initial value)   |
|            | Time reference message with Next_is_Gap has not been received in time-trigger mode message error has not occurred in test mode.   |
|            | [Clearing condition] Writing '1' to IRR13   |
| 1          | [Setting condition]   |
|            | Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-<br>trigger mode.time reference message with Next_is_Gap has been received<br>in time-trigger mode message error has occurred in test mode |

#### (5) Cycle Counter Register (CCR)

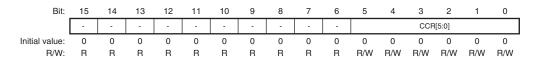
This register is a 6-bit read/write register. Its purpose is to store the number of the basic cycle for Time -Triggered Transmissions. Its value is updated in different fashions depending if RCAN-TL1 is programmed to work as a potential time master or as a time slave. If RCAN-TL1 is working as (potential) time master, CCR is:

- Incremented by one every time the cycle time (CYCTR) matches to Tx-Trigger Time of Mailbox-30 or
- Overwritten with the value contained in MSG\_DATA\_0[5:0] of Mailbox 31 when a valid reference message is received.

If RCAN-TL1 is working as a time slave, CCR is only overwritten with the value of MSG\_DATA\_0[5:0] of Mailbox 31 when a valid reference message is received.

If CMAX = 3'111, CCR is always H'0000.

• CCR (Address = H'08A)



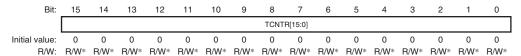
Bits 15 to 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 5 to 0 — Cycle Counter Register (CCR): Indicates the number of the current Base Cycle of the matrix cycle for Timer Triggered transmission.

## (6) Timer Counter Register (TCNTR)

This is a 16-bit read/write register that allows the CPU to monitor and modify the value of the Free Running Timer Counter. When the Timer meets TCMR0 (Timer Compare Match Register 0) + TTCR0 [6] is set to '1', the TCNTR is cleared to H'0000 and starts running again. In Time-Trigger mode, this timer can be used as Local Time and TTCR0[6] has to be cleared to work as a free running timer.

- Notes: 1. It is possible to write into this register only when it is enabled by the bit 15 in TTCR0. If TTCR0 bit15 = 0, TCNTR is always H'0000.
  - 2. There could be a delay of a few clock cycles between the enabling of the timer and the moment where TCNTR starts incrementing. This is caused by the internal logic used for the pre-scaler.
- TCNTR (Address = H'08C)



Note: \* The register can be written only when enabled in TTCR0[15]. Write operation is not allowed in Time Trigger mode (i.e. CMAX is not 3'b111).

Bit 15 to 0 — Indicate the value of the Free Running Timer.

# (7) Cycle Time register (CYCTR)

This register is a 16-bit read-only register. This register shows Cycle Time = Local Time (TCNTR) - Reference\_Mark (RFMK). In ET mode this register is the exact copy of TCNTR as RFMK is always fixed to zero.

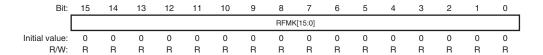
• CYCTR (Address = H'090)

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

## (8) Reference Mark Register (RFMK)

This register is a 16-bit read-only register. The purpose of this register is to capture Local Time (TCNTR) at SOF of the reference message when the message is received or transmitted successfully. In ET mode this register is not used and it is always cleared to zero.

• RFMK (Address = H'094)



Bit 15 to 0 — Reference Mark Register (RFMK): Indicates the value of TCNTR at SOF of time reference message.

## (9) Timer Compare Match Registers (TCMR0, TCMR1, TCMR2)

These three registers are 16-bit read/write registers and are capable of generating interrupt signals, clearing-setting the Timer value (only supported by TCMR0) or clear the transmission messages in the queue (only supported by TCMR2). TCMR0 is compared with TCNTR, however, TCMR1 and TCMR2 are compared with CYCTR.

The value used for the compare can be configured independently for each register. In order to set flags, TTCR0 bit 12-10 needs to be set.

In Time-Trigger mode, TTCR0 bit6 has to be cleared by software to prevent TCNTR from being cleared.

TMCR0 is for Init\_Watch\_Trigger, and TCMR2 is for Watch\_Trigger.

## Interrupt:

The interrupts are flagged by the Bit11, Bit15 and 14 in the IRR accordingly when a Compare Match occurs, and setting these bits can be enabled by Bit12, Bit11, Bit10 in TTCR0. The generation of interrupt signals itself can be prevented by the Bit11, Bit15 and Bit14 in the IMR. When a Compare Match occurs and the IRR11 (or IRR15 or IRR14) is set, the Bit3 or Bit2 or Bit1 in the TSR (Timer Status Register) is also set. Clearing the IRR bit also clears the corresponding bit of TSR.

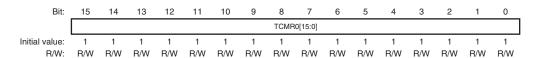
#### Timer Clear-Set:

The Timer value can only be cleared when a Compare Match occurs if it is enabled by the Bit6 in the TTCR0. TCMR1 and TCMR2 do not have this function.

## Cancellation of the messages in the transmission queue:

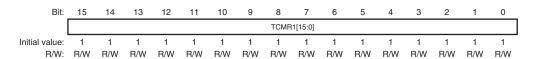
The messages in the transmission queue can only be cleared by the TCMR2 through setting TXCR when a Compare Match occurs while RCAN-TL1 is not in the halt status. TCMR1 and TCMR0 do not have this function.

TCMR0 (Address = H'098)



Bit 15 to 0 — Timer Compare Match Register (TCMR0): Indicates the value of TCNTR when compare match occurs.

• TCMR1 (Address = H'09C)



Bit 15 to 0 — Timer Compare Match Register (TCMR1): Indicates the value of CYCTR when compare match occurs.

• TCMR2 (Address = H'0A0)

| Bit:           | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|-------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|                | TCMR2[15:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value: | 1           | 1  | 1  | 1  | 1  | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bit 15 to 0 — Timer Compare Match Register (TCMR2): Indicates the value of CYCTR when compare match occurs.

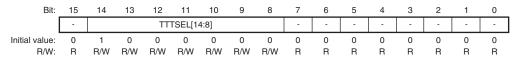
## (10) Tx-Trigger Time Selection Register (TTTSEL)

This register is a 16-bit read/write register and specifies the Tx-Trigger Time waiting for compare match with Cycle Time. Only one bit is allowed to be set. Please don't set more bits than one, or clear all bits.

This register may only be modified during configuration mode. The modification algorithm is shown in Figure 18.13.

Please note that this register is only indented for test and diagnosis. When not in test mode, this register must not be written to and the returned value is not guaranteed.

• TTTSEL (Address = H'0A4)



Note: Only one bit is allowed to be set.

Bit 15: Reserved. The written value should always be '0' and the returned value is '0'.

**Bit 14 to 8** — Specifies the Tx-Trigger Time waiting for compare match with CYCTR The bit 14 to 8 corresponds to Mailbox-30 to 24, respectively.

Bits 7 to 0: Reserved. The written value should always be '0' and the returned value is '0'.

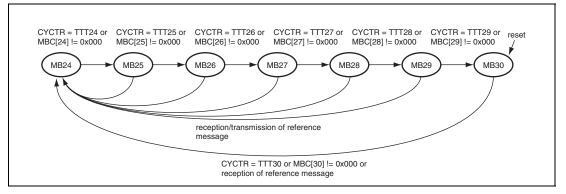


Figure 18.13 TTTSEL modification algorithm

# 18.4 Application Note

## **18.4.1** Test Mode Settings

The RCAN-TL1 has various test modes. The register TST[2:0] (MCR[10:8]) is used to select the RCAN-TL1 test mode. The default (initialized) settings allow RCAN-TL1 to operate in Normal mode. The following table is examples for test modes.

Test Mode can be selected only while in configuration mode. The user must then exit the configuration mode (ensuring BCR0/BCR1 is set) in order to run the selected test mode.

| Bit10:<br>TST2 | Bit9:<br>TST1 | Bit8:<br>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-TL1 operates in the normal mode.

Listen-Only Mode: ISO-11898 requires this mode for baud rate detection. The Error Counters

are cleared and disabled so that the TEC/REC does not increase the values, and the CTxn (n = A, B, C) Output is disabled so that RCAN-TL1 does not

generate error frames or acknowledgment bits. IRR13 is set when a

message error occurs.

Self Test Mode 1: RCAN-TL1 generates its own Acknowledge bit, and can store its own

messages into a reception mailbox (if required). The CRxn/CTxn (n = A,

B, C) pins must be connected to the CAN bus.

Self Test Mode 2: RCAN-TL1 generates its own Acknowledge bit, and can store its own

messages into a reception mailbox (if required). The CRxn/CTxn (n = A, B, C) pins do not need to be connected to the CAN bus or any external devices, as the internal CTxn (n = A, B, C) is looped back to the internal CRxn (n = A, B, C). CTxn (n = A, B, C) pin outputs only recessive bits and

CRxn (n = A, B, C) pin is disabled.

Write Error Counter: TEC/REC can be written in this mode. RCAN-TL1 can be forced to

become an Error Passive mode by writing a value greater than 127 into the Error Counters. The value written into TEC is used to write into REC, so only the same value can be set to these registers. Similarly, RCAN-TL1 can be forced to become an Error Warning by writing a value greater than

95 into them.

RCAN-TL1 needs to be in Halt Mode when writing into TEC/REC (MCR1 must be "1" when writing to the Error Counter). Furthermore this

test mode needs to be exited prior to leaving Halt mode.

Error Passive Mode: RCAN-TL1 can be forced to enter Error Passive mode.

Note: The REC will not be modified by implementing this Mode. However, once running in Error Passive Mode, the REC will increase normally should errors be received. In this Mode, RCAN-TL1 will enter BusOff if TEC reaches 256 (Dec). However when this mode is used RCAN-TL1 will not be able to become Error Active. Consequently, at the end of the Bus Off recovery sequence, RCAN-TL1 will move to Error

Passive and not to Error Active.

When message error occurs, IRR13 is set in all test modes.

#### **Configuration of RCAN-TL1** 18.4.2

RCAN-TL1 is considered in configuration mode or after a H/W (Power On Reset)/S/W (MCR[0]) reset or when in Halt mode. In both conditions RCAN-TL1 cannot join the CAN Bus activity and configuration changes have no impact on the traffic on the CAN Bus.

#### After a Reset request

The following sequence must be implemented to configure the RCAN-TL1 after (S/W or H/W) reset. After reset, all the registers are initialized, therefore, RCAN-TL1 needs to be configured before joining the CAN bus activity. Please read the notes carefully.

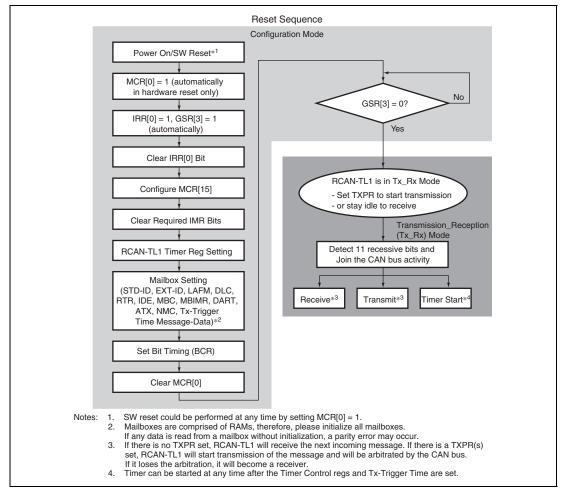


Figure 18.14 Reset Sequence

#### Halt mode

When RCAN-TL1 is in Halt mode, it cannot take part to the CAN bus activity. Consequently the user can modify all the requested registers without influencing existing traffic on the CAN Bus. It is important for this that the user waits for the RCAN-TL1 to be in halt mode before to modify the requested registers—note that the transition to Halt Mode is not always immediate (transition will occurs when the CAN Bus is idle or in intermission). After RCAN-TL1 transit to Halt Mode, GSR4 is set.

Once the configuration is completed the Halt request needs to be released. RCAN-TL1 will join CAN Bus activity after the detection of 11 recessive bits on the CAN Bus.

#### Sleep mode

When RCAN-TL1 is in sleep mode the clock for the main blocks of the IP is stopped in order to reduce power consumption. Only the following user registers are clocked and can be accessed: MCR, GSR, IRR and IMR. Interrupt related to transmission (TXACK and ABACK) and reception (RXPR and RFPR) cannot be cleared when in sleep mode (as TXACK, ABACK, RXPR and RFPR are not accessible) and must to be cleared beforehand.

The following diagram shows the flow to follow to move RCAN-TL1 into sleep mode.

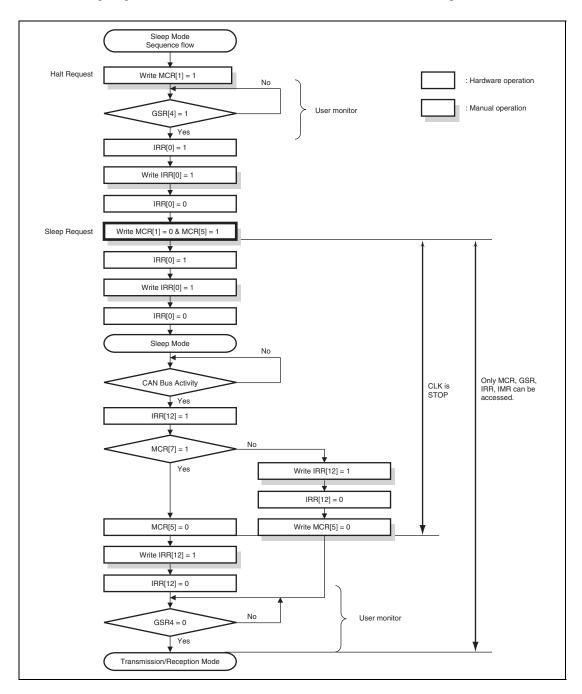


Figure 18.15 shows allowed state transitions.

- Please don't set MCR5 (Sleep Mode) without entering Halt Mode.
- After MCR1 is set, please don't clear it before GSR4 is set and RCAN-TL1 enters Halt Mode.

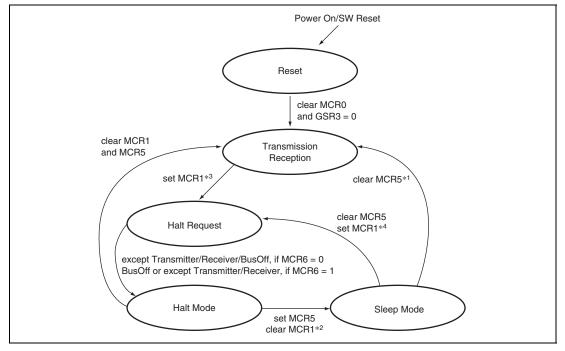


Figure 18.15 Halt Mode/Sleep Mode

Notes: 1. MCR5 can be cleared by automatically by detecting a dominant bit on the CAN Bus if MCR7 is set or by writing '0'.

- 2. MCR1 is cleared in SW. Clearing MCR1 and setting MCR5 have to be carried out by the same instruction.
- 3. MCR1 must not be cleared in SW, before GSR4 is set. MCR1 can be set automatically in HW when RCAN-TL1 moves to Bus Off and MCR14 and MCR6 are both set.
- 4. When MCR5 is cleared and MCR1 is set at the same time, RCAN-TL1 moves to Halt Request. Right after that, it moves to Halt Mode with no reception/transmission.

The following table shows conditions to access registers.

#### **RCAN-TL1 Registers**

| Status Mode                               | MCR<br>GSR |     | BCR  | MBIMR<br>timer<br>TT_register | Flag_<br>register | Mailbox<br>(ctrl0, L |       | Mailbox<br>(data) | Mailbox<br>(ctrl1) | Mailbox<br>Trigger<br>Time TT<br>control |
|---|------------|-----|------|-------------------------------|-------------------|----------------------|-------|-------------------|--------------------|--|
| Reset                                     | yes        | yes | yes  | yes                           | yes               | yes                  |       | yes               | yes                | yes                                      |
| Transmission<br>Reception<br>Halt Request | yes        | yes | no*1 | yes                           | yes               | no*1                 | yes*2 | yes*2             | no*1 yes*2         | yes* <sup>2</sup>                        |
| Halt                                      | yes        | yes | no*1 | yes                           | yes               | yes                  |       | yes               | yes                | yes                                      |
| Sleep                                     | yes        | yes | no   | no                            | no                | no                   |       | no                | no                 | no                                       |

Notes: 1. No hardware protection.

2. When TXPR is not set.

## 18.4.3 Message Transmission Sequence

#### Message Transmission Request

The following sequence is an example to transmit a CAN frame onto the bus. As described in the previous register section, please note that IRR8 is set when one of the TXACK or ABACK bits is set, meaning one of the Mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, the GSR2 means that there is currently no transmission request made (No TXPR flags set).

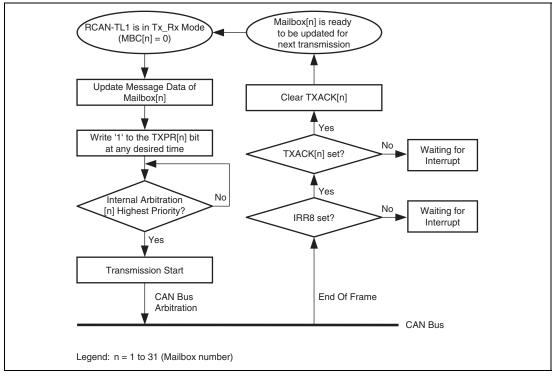


Figure 18.16 Transmission request

#### • Internal Arbitration for transmission

The following diagram explains how RCAN-TL1 manages to schedule transmission-requested messages in the correct order based on the CAN identifier. 'Internal arbitration' picks up the highest priority message amongst transmit-requested messages.

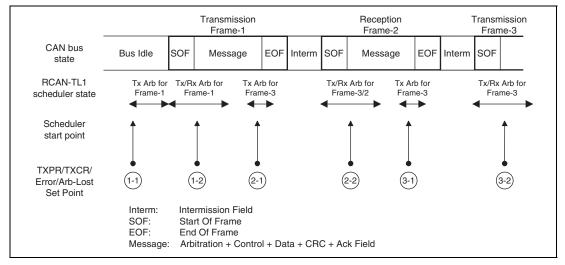


Figure 18.17 Internal Arbitration for transmission

The RCAN-TL1 has two state machines. One is for transmission, and the other is for reception.

- 1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.
- 1-2: Operations for both transmission and reception starts at SOF. Since there is no reception frame, RCAN-TL1 becomes transmitter.
- 2-1: At crc delimiter, internal arbitration to search next message transmitted starts.
- 2-2: Operations for both transmission and reception starts at SOF. Because of a reception frame with higher priority, RCAN-TL1 becomes receiver. Therefore, Reception is carried out instead of transmitting Frame-3.
- 3-1: At crc delimiter, internal arbitration to search next message transmitted starts.
- 3-2: Operations for both transmission and reception starts at SOF. Since a transmission frame has higher priority than reception one, RCAN-TL1 becomes transmitter.

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

As the arbitration for transmission is performed at CRC delimiter, in case a remote frame request is received into a Mailbox with ATX = 1 the answer can join the arbitration for transmission only at the following Bus Idle, CRC delimiter or Error Delimiter.

Depending on the status of the CAN bus, following the assertion of the TXCR, the corresponding Message abortion can be handled with a delay of maximum 1 CAN Frame.

#### (1) Time Triggered Transmission

RCAN-TL1 offers a H/W support to perform communication in Time Trigger mode in line with the emerging ISO-11898-4 Level 1 Specification.

This section reports the basic procedures to use this mode.

## • Setting Time Trigger Mode

In order to set up the time trigger mode the following settings need to be used.

- CMAX in CMAX\_TEW must be programmed to a value different from 3'b111.
- Bit 15 in TTCR0 has to be set, to start TCNTR.

Local Time

- Bit 6 in TTCR0 has to be cleared to prevent TCNTR from being cleared after a match.
- DART in Mailboxes used for time-triggered transmission cannot be used, since for Time Triggered Mailboxes, TXPR is not cleared to support periodic transmission.

## • Roles of Registers

TCNTP

The user registers of RCAN-TL1 can be used to handle the main functions requested by the TTCAN standard.

| ICNIK            | Local Time   |  |
|------------------|--|--|
| RFMK             | Ref_Mark   |  |
| CYCTR            | Cycle Time = TCNTR - RFMK  |  |
| RFTROFF          | Ref_Trigger_Offset for Mailbox-30  |  |
| Mailbox-31       | Mailbox dedicated to the reception of time reference message   |  |
| Mailbox-30       | Mailbox dedicated to the transmission of time reference message when working as a potential time master                                |  |
| Mailbox-29 to 24 | Mailboxes supporting time-triggered transmission   |  |
| Mailbox-23 to 16 | Mailboxes supporting reception without timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission)        |  |
| Mailbox-15 to 0  | Mailboxes supporting reception with timestamp timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission) |  |
| Tx-Trigger Time  | Time_Mark to specify when a message should be transmitted  |  |

| CMAX       | Specifies the maximum  | number of basic cycles     | when working as potential |
|------------|------------------------|----------------------------|---------------------------|
| C1717 17 1 | opecifies the maximum. | indifficer of ouble cycles | when working as potential |

time master

TEW Specify the width of Tx\_Enable

TCMR0 Init Watch Trigger (compare match with Local Time)

TCMR1 Compare match with Cycle Time to monitor users-specified events

TCMR2 Watch\_Trigger (compare match with Cycle Time). This can be programmed

to abort all pending transmissions

TTW Specifies the attribute of a time window used for transmission

TTTSEL Specifies the next Mailbox waiting for transmission

#### Time Master/Time Slave

RCAN-TL1 can be programmed to work as a potential time master of the network or as a time slave. The following table shows the settings and the operation automatically performed by RCAN-TL1 in each mode.

| mode        | requested setting                      | function   |  |  |  |
|-------------|--|--|--|--|--|
| Time Slave  | TXPR[30] = 0<br>&<br>MBC[30]!= 3'b000  | TCNTR is sampled at each SOF detected on the CAN Bus and stored into an internal register. When a valid Time Reference Message is received into Mailbox-31 the value of TCNTR (stored at the SOF) is copied into Ref_Mark. |  |  |  |
|             | &<br>CMAX!= 3'b111                     | CCR embedded in the received Reference Message is copied to CCR.   |  |  |  |
|             | &<br>MBC[31] = 3'b011                  | If Next_is_Gap = 1, IRR13 is set.  |  |  |  |
| (Potential) | TXPR[30] = 1                           | Two cases are covered:   |  |  |  |
| Time Master | &<br>MBC[30] = 3'b000<br>&             | (1) When a valid Time Reference message is received in Mailbox-31 the value of TCNTR stored into an interna register at the SOF is copied into Ref_Mark.   |  |  |  |
|             | DLC[30] > 0                            | CCR embedded in the received Reference Message is copied to CCR.   |  |  |  |
|             | &                                      | If Next_is_Gap = 1, IRR13 is set.  |  |  |  |
|             | CMAX!= 3'b111<br>&<br>MBC[31] = 3'b011 | (2) When a Time Reference message is transmitted from<br>Mailbox-30 the value of TCNTR stored into an internal<br>register at the SOF is copied into Ref_Mark.   |  |  |  |
|             | = 0.011                                | CCR is incremented when TTT of Mailbox-30 matches with CYCTR.  |  |  |  |
|             |  | CCR is embedded into the first data byte of the time reference message   |  |  |  |
|             |  | { Data0[7:6], CCR[5:0] } .   |  |  |  |

# • Setting Tx-Trigger Time

The Tx-Trigger Time(TTT) must be set in ascending order shown below, and the difference between them has to satisfy the following expressions. TEW in the following expressions is the register value.

TTT (Mailbox-24) < TTT (Mailbox-25) < TTT (Mailbox-26) < TTT (Mailbox-27) < TTT (Mailbox-28) < TTT (Mailbox-29) < TTT (Mailbox-30)

and

TTT (Mailbox-n-1) > TEW + the maximum frame length + 9 n = 25 to 30

TTT (Mailbox-24) to TTT (Mailbox-29) correspond to Time\_Marks, and TTT (Mailbox-30) corresponds to Time\_Ref showing the length of a basic cycle, respectively when working as potential time master.

The above limitation is not applied to mailboxes which are not set as time-triggered transmission.

**Important:** Because of limitation on setting Tx-Trigger Time, only one Mailbox can be assigned to one time window.

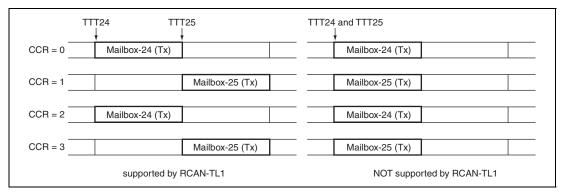


Figure 18.18 Limitation on Tx-Trigger Time

The value of TCMR2 as Watch\_Trigger has to be larger than TTT(Mailbox-30), which shows the length of a basic cycle.

Figure 18.19 and Figure 18.20 show examples of configurations for (Potential) Time Master and Time Slave. "L" in diagrams shows the length in time of the time reference messages.

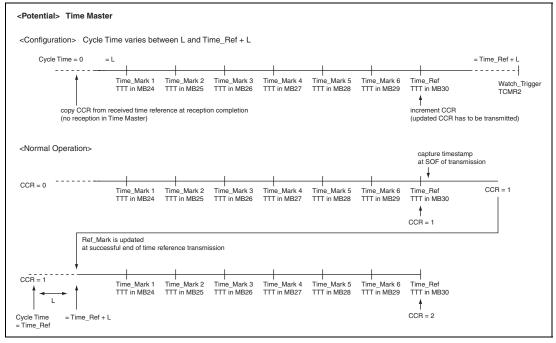


Figure 18.19 (Potential) Time Master

Apr 01, 2014

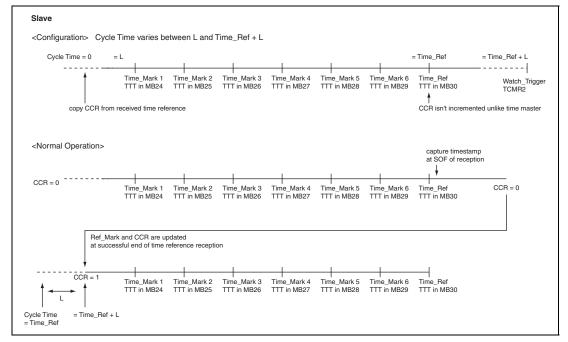


Figure 18.20 Time Slave

• Function to be implemented by software

Some of the TTCAN functions need to be implemented in software. The main details are reported hereafter. Please refer to ISO-11898-4 for more details.

— Change from Init\_Watch\_Trigger to Watch\_Trigger RCAN-TL1 offers the two registers TCMR0 and TCMR2 as H/W support for Init\_Watch\_Trigger and Watch\_Trigger respectively. The SW is requested to enable TCMR0 and disable TCMR2 up to the first reference message is detected on the CAN Bus and then disable TCMR0 and enable TCMR2.- Schedule Synchronization state machine.

Only reception of Next\_is\_Gap interrupt is supported. The application needs to take care of stopping all transmission at the end of the current basic cycle by setting the related TXCR flags.Master-Slave Mode control.

Only automatic cycle time synchronization and CCR increment is supported.

Message status count
 Software has to count scheduling errors for periodic messages in exclusive windows.

• Message Transmission Request for Time Triggered communication

When the Time Triggered mode is used communications must fulfils the ISO11898-4 requirements.

The following procedure should be used.

- Send RCAN-TL1 to reset or halt mode
- Set TCMR0 to the Init\_Watch\_Trigger (0xFFFF)
- Enable TCMR0 compare match setting bit 10 of TTCR0
- Set TCMR2 to the specified Watch\_Trigger value
- Keep TCMR2 compare match disabled by keeping cleared the bit 12 of TTCR0
- Set CMAX to the requested value (different from 111 bin)
- Set TEW to the requested value
- Configure the necessary Mailboxes for Time Trigger transmission and reception
- Set LAFM for the 3 LSBs of Mailbox 31
- Configure MCR, BCR1 and BCR0 to the requested values
- If working as a potential time master:
  - Set RFTROFF to the requested Init\_Ref\_Offset value
  - Set TXPR for Mailbox 30
  - Write H'4000 into TTTSEL.
- Enable the TCNTR timer through the bit 15 of TTCR0
- Move to Transmission Reception mode
- Wait for the reception or transmission of a valid reference message or for TCMR0 match
- If the local time reaches the value of TCMR0 the Init\_Watch\_Trigger is reached and the application needs to set TXCR for Mailbox 30 and start again
- If the reference message is transmitted (TXACK[30] is set) set RFTROFF to zero
- If a valid reference message is received (RXPR[31] is set) then:
  - If 3 LSBs of ID of Mailbox 31 have high priority than the 3 LSBs of Mailbox 30 (if working as potential time master) keep RFTROFF to Init\_Ref\_Offset
  - If 3 LSBs of ID of Mailbox 31 have lower priority than the 3 LSBs of Mailbox 30 (if working as potential time master) decrement by 1 the value in RFTROFF
- Disable TCMR0 compare match by clearing bit 10 of TTCR0
- Enable TCMR2 compare match by setting bit 12 of TTCR0
- Only after two reference messages have been detected on the CAN Bus (transmitted or received) can the application set TXPR for the other Time Triggered Mailboxes.

If, at any time, a reference message cannot be detected on the CAN Bus, and the cycle time CYCTR reaches TCMR2, RCAN-TL1 automatically aborts all pending transmissions (including the Reference Message).

The following is the sequence to request further transmission in Time Triggered mode.

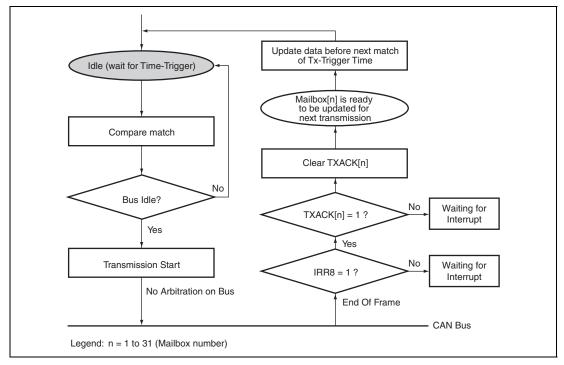


Figure 18.21 Message transmission request

S/W has to ensure that a message is updated before a Tx trigger for transmission occurs.

When the CYCTR reaches to TTT (Tx-Trigger Time) of a Mailbox and CCR matches with the programmed cycle for transmission, RCAN-TL1 immediately transfers the message into the Tx buffer. At this point, RCAN-TL1 will attempt a transmission within the specified Time Enable Window. If RCAN-TL1 misses this time slot, it will suspend the transmission request up to the next Tx Trigger, keeping the corresponding TXPR bit set to '1' if the transmission is periodic (Mailbox-24 to 30). There are three factors that may cause RCAN-TL1 to miss the time slot –

- 1. The CAN bus currently used
- 2. An error on the CAN bus during the time triggered message transmission
- 3. Arbitration loss during the time triggered message transmission

In case of Merged Arbitrating Window the slot for transmission goes from the Tx\_Trig of the Mailbox opening the Window (TTW = 10 bin) to the end to the TEW of the Mailbox closing the Window (TTW = 11 bin). The TXPR can be modified at any time. RCAN-TL1 ensures the transmission of Time Triggered messages is always scheduled correctly. However, in order to guarantee the correct schedule, there are some important rules that are:

- TTT (Tx Trigger Time) can be modified during configuration mode.
- TTT cannot be set outside the range of Time\_Ref, which specifies the length of basic cycle. This could cause a scheduling problem.
- TXPR is not automatically cleared for periodic transmission. If a periodic transmission needs to be cancelled, the corresponding TXCR bit needs to be set by the application.

#### Example of Time Triggered System

The following diagram shows a simple example of how time trigger system works using RCAN-TL1 in time slave mode.

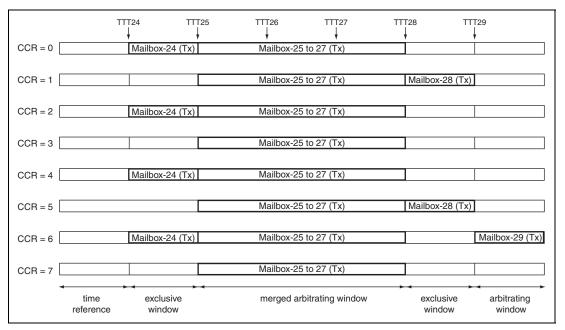


Figure 18.22 Example of Time trigger system as Time Slave

The following settings were used in the above example:

|            | rep_factor<br>(register) | Offset    | TTW[1:0] | MBC[2:0] |
|------------|--------------------------|-----------|----------|----------|
| Mailbox-24 | 3'b001                   | 6'b000000 | 2'b00    | 3'b000   |
| Mailbox-25 | 3'b000                   | 6'b000000 | 2'b10    | 3'b000   |
| Mailbox-26 | 3'b000                   | 6'b000000 | 2'b10    | 3'b000   |
| Mailbox-27 | 3'b000                   | 6'b000000 | 2'b11    | 3'b000   |
| Mailbox-28 | 3'b010                   | 6'b000001 | 2'b00    | 3'b000   |
| Mailbox-29 | 3'b011                   | 6'b000110 | 2'b01    | 3'b000   |
| Mailbox-30 | _                        | _         | _        | 3'b111   |
| Mailbox-31 | _                        | _         | _        | 3'b011   |

CMAX = 3'b011, TXPR[30] = 0

During merged arbitrating window, request by time-triggered transmission is served in the way of FCFS (First Come First Served). For example, if Mailbox-25 cannot be transmitted between Tx-Trigger Time 25 (TTT25) and TTT26, Mailbox-25 has higher priority than Mailbox-26 between TTT26 and 28.

MBC needs to be set into 3'b111, in order to disable time-triggered transmission. If RCAN-TL1 is Time Master, MBC[30] has to be 3'b000 and time reference window is automatically recognized as arbitrating window.

## • Timer Operation

Figure 18.23 shows the timing diagram of the timer. By setting Tx-Trigger Time = n, time trigger transmission starts between CYCTR = n + 2 and CYCTR = n + 3.

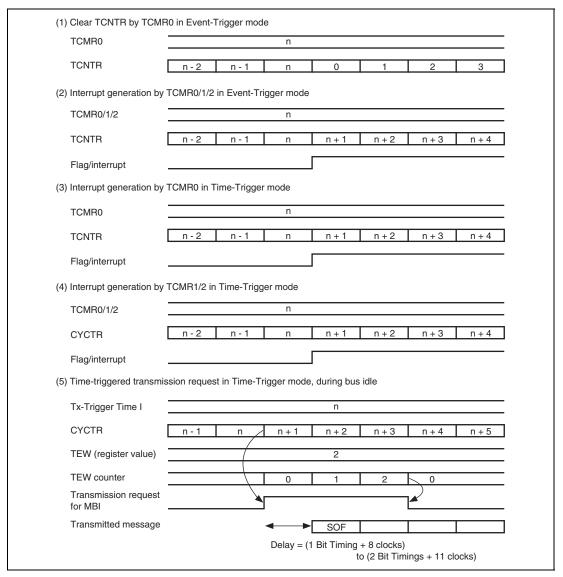


Figure 18.23 Timing Diagram of Timer

During merged arbitrating window, event-trigger transmission is served after completion of time-triggered transmission. For example, If transmission of Mailbox-25 is completed and CYCTR doesn't reach TTT26, event-trigger transmission starts based on message transmission priority specified by MCR2. TXPR of time-triggered transmission is not cleared after transmission completion, however, that of event-triggered transmission is cleared.

that in the case that the TXPR is not set for the Mailbox which is assigned to close the Merged Arbitrating Window (MAW), then the MAW will still be closed (at the end of the TEW following the TTT of the assigned Mailbox.

Please refer to section 18.3.2, Mailbox Structure.

#### 18.4.4 Message Receive Sequence

The diagram below shows the message receive sequence.

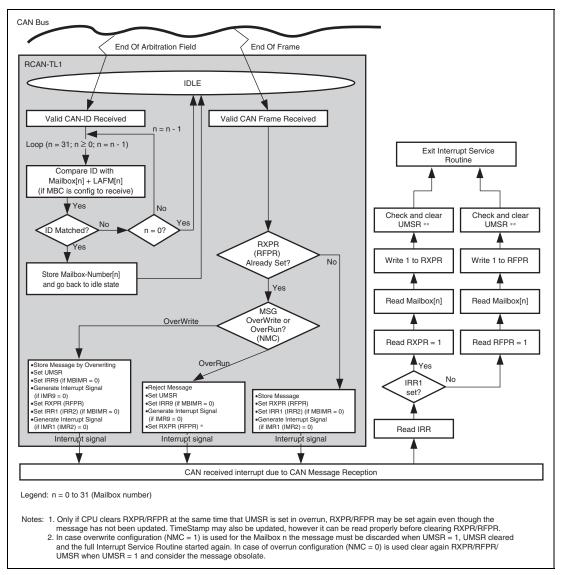


Figure 18.24 Message receive sequence

When RCAN-TL1 recognises the end of the Arbitration field while receiving a message, it starts comparing the received identifier to the identifiers set in the Mailboxes, starting from Mailbox-31 down to Mailbox-0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox-31 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-30 (if configured as receive). Once RCAN-TL1 finds a matching identifier, it stores the number of Mailbox-[n] into an internal buffer, stops the search, and goes back to idle state, waiting for the EndOfFrame (EOF) to come. When the 6th bit of EOF is notified by the CAN Interface logic, the received message is written or abandoned, depending on the NMC bit. No modification of configuration during communication is allowed. Entering Halt Mode is one of ways to modify configuration. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of a received message matches to ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

With regards to the reception of data and remote frames described in the above flow diagram the clearing of the UMSR flag after the reading of IRR is to detect situations where a message is overwritten by a new incoming message stored in the same mailbox (if its NMC = 1) while the interrupt service routine is running. If during the final check of UMSR a overwrite condition is detected the message needs to be discarded and read again.

In case UMSR is set and the Mailbox is configured for overrun (NMC = 0) the message is still valid, however it is obsolete as it is not reflecting the latest message monitored on the CAN Bus.

Please access the full Mailbox content before clearing the related RXPR/RFPR flag.

Please note that in the case a received remote frame is overwritten by a data frame, both the remote frame receive interrupt (IRR2) and data frame received interrupt (IRR1) and also the Receive Flags (RXPR and RFPR) are set. In an analogous way, the overwriting of a data frame by a remote frame, leads to setting both IRR2 and IRR1.

When a message is received and stored into a Mailbox all the fields of the data not received are stored as zero. The same applies when a standard frame is received. The extended identifier part (EXTID[17:0]) is written as zero.

#### 18.4.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

- Change configuration of transmit box Two cases are possible.
  - Change of ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART
     This change is possible only when MBC = 3'b000. Confirm that the corresponding TXPR is not set. The configuration (except MBC bit) can be changed at any time.
  - Change from transmit to receive configuration (MBC)
    Confirm that the corresponding TXPR is not set. The configuration can be changed only in Halt or reset state. Please note that it might take longer for RCAN-TL1 to transit to halt state if it is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-TL1 will not be able to receive/transmit messages during the Halt state.
    - In case RCAN-TL1 is in the Bus Off state the transition to halt state depends on the configuration of the bit 6 of MCR and also bit and 14 of MCR.
- Change configuration (ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART, MBC) of receiver box or Change receiver box to transmitter box
   The configuration can be changed only in Halt Mode.
  - RCAN-TL1 will not lose a message if the message is currently on the CAN bus and RCAN-TL1 is a receiver. RCAN-TL1 will be moving into Halt Mode after completing the current reception. Please note that it might take longer if RCAN-TL1 is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-TL1 will not be able to receive/transmit messages during the Halt Mode.

In case RCAN-TL1 is in the Bus Off state the transition to halt mode depends on the configuration of the bit 6 and 14 of MCR.

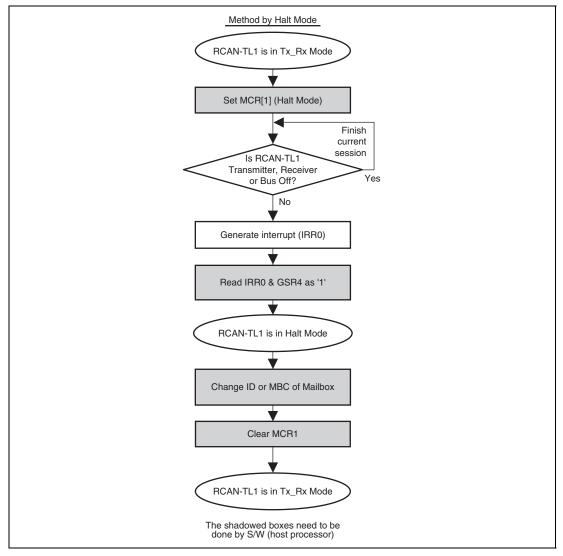


Figure 18.25 Change ID of receive box or Change receive box to transmit box

## 18.5 Parity Detection

The RCAN-TL1 adds parity bits when data are written to a mailbox (RAM) and performs parity error detection when data are read from a mailbox. Interrupt generation upon parity error detection can be selected by setting the MBECR register.

- Parity addition: One parity bit is added to every 8 bits of data.
- Parity detection: Parity detection is performed in 32-bit units.

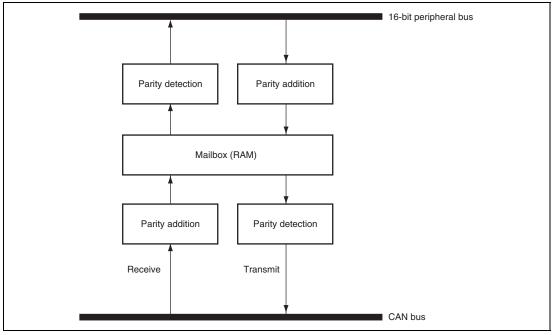


Figure 18.26 Parity Addition and Detection

## 18.6 Interrupt Sources

Table 18.2 lists the RCAN-TL1 interrupt sources. These sources can be masked. Masking is implemented using the mailbox interrupt mask registers (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 8, Interrupt Controller (INTC).

**Table 18.2 RCAN-TL1-n\*1 Interrupt Sources** 

| Interrupt                         | Description   | Interrupt<br>Flag | DMAC<br>Activation | A-DMAC<br>Activation |
|-----------------------------------|---|-------------------|--------------------|----------------------|
| ERSn*1                            | Error Passive Mode (TEC $\geq$ 128 or REC $\geq$ 128)   | IRR5              | Not possible       | Not possible         |
|                                   | Bus Off (TEC ≥ 256)/Bus Off recovery                    | IRR6              | •                  |                      |
|                                   | Error warning (TEC ≥ 96)                                | IRR3              | •                  |                      |
|                                   | Error warning (REC ≥ 96)                                | IRR4              | •                  |                      |
| OVRn*1                            | Reset/halt/CAN sleep transition                         | IRR0              | •                  |                      |
|                                   | Overload frame transmission                             | IRR7              | •                  |                      |
|                                   | Unread message overwrite (overrun)                      | IRR9              | •                  |                      |
|                                   | Start of new system matrix                              | IRR10             | •                  |                      |
|                                   | TCMR2 compare match                                     | IRR11             | •                  |                      |
|                                   | Bus activity while in sleep mode                        | IRR12             | •                  |                      |
|                                   | Timer overrun/Next_is_Gap reception/message error       | IRR13             |                    |                      |
|                                   | TCMR0 compare match                                     | IRR14             | •                  |                      |
|                                   | TCMR1 compare match                                     | IRR15             | •                  |                      |
|                                   | Data frame reception                                    | IRR1*3            | Possible*4         | Possible*5           |
| RMn1* <sup>1</sup> * <sup>2</sup> | Remote frame reception                                  | IRR2*3            | •                  |                      |
| SLEn*1                            | Message transmission/transmission disabled (slot empty) | IRR8              | Not possible       | Not possible         |
| MBEn*1                            | Message buffer error                                    | MBEF              | Not possible       | Not possible         |

Notes: 1. n = A. B. C

- 2. RM0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1 is an interrupt generated by the remote request pending flag for mailboxes 1 to 31 (RFPR1/RFPR0) or the data frame receive flag for mailboxes 1 to 31 (RXPR1/RXPR0).
- 3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 31, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 31.
- 4. The DMAC is activated only by an RMn0 interrupt.
- 5. The A-DMAC can be activated by an interrupt in any mailboxes.

#### 18.7 DMAC Interface

The DMAC can be activated by the reception of a message in RCAN-TL1 mailbox 0. When DMAC transfer ends after DMAC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-TL1 cannot be sent to the CPU in this case. Figure 18.27 shows a DMAC transfer flowchart.

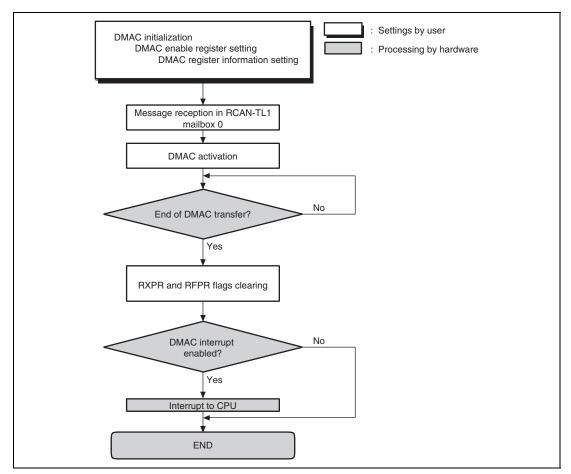


Figure 18.27 DMAC Transfer Flowchart

#### CAN Bus Interface 18.8

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. Figure 18.28 shows a sample connection diagram.

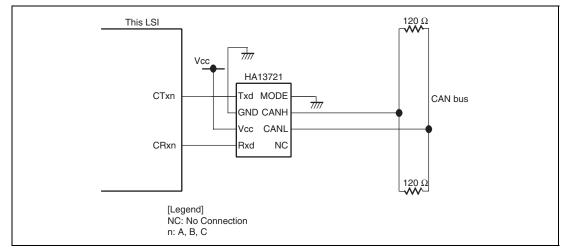


Figure 18.28 High-Speed CAN Interface Using HA13721

#### **A-DMAC** Interface 18.9

The A-DMAC can be activated by reception of a message in any Mailboxes in any of three channels (RCAN-TL1-A, RCAN-TL1-B, and RCAN-TL1-C).

When a message is received in a Mailbox after setting is completed, the A-DMAC is activated. The A-DMAC transfers the message data from the Mailbox to the RAM area and the flag in RXPR or RFPR is cleared to 0. At this time, an interrupt to the CPU is not generated by the RCAN-TL1 because it is masked by IMR.

For details on the setting of the A-DMAC, see section 12, Automotive Direct Memory Access Controller (A-DMAC).

## 18.10 Setting I/O Ports for RCAN-TL1

The I/O ports for the RCAN-TL1 must be specified before or during the configuration mode. For details on the settings of I/O ports, see section 22, Pin Function Controller (PFC). Three methods are available using three channels of the RCAN-TL1 in this LSI.

- Using RCAN-TL1 as a 3-channel module (channels A, B, and C) Each channel has 32 Mailboxes.
- Using RCAN-TL1 as a 2-channel module (channels A and B functioning as a single channel) Channels A and B as a single channel has 64 Mailboxes and channel C has 32 Mailboxes.
- Using RCAN-TL1 as a 1-channel module (channels A, B, and C functioning as a single channel)

Channels A, B, and C as a single channel has 96 Mailboxes.

When the second or third method is used, see section 18.11.1, Notes on Port Setting for Multiple Channels Used as Single Channel.

Figures 18.29 to 18.31 show connection examples for individual port settings.

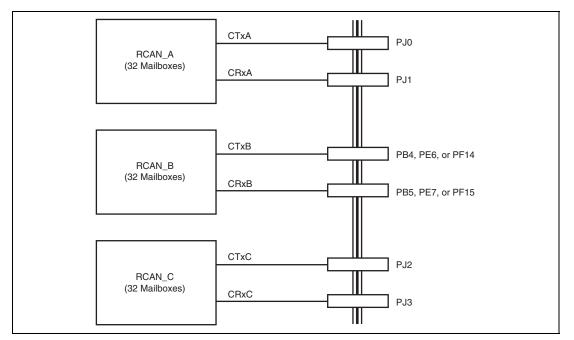


Figure 18.29 Connection Example when Using RCAN-TL1 as 3-Channel Module (32 Mailboxes × 3 Channels)

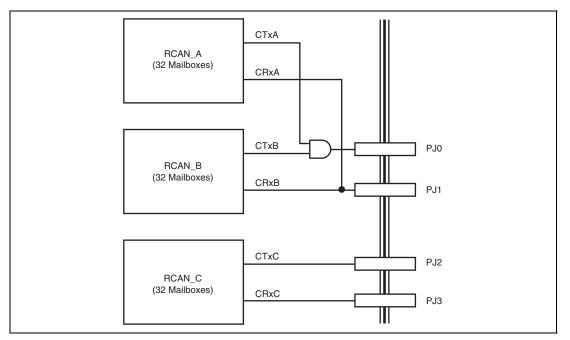


Figure 18.30 Connection Example when Using RCAN-TL1 as 2-Channel Module (64 Mailboxes × 1 Channel and 32 Mailboxes × 1 Channel)

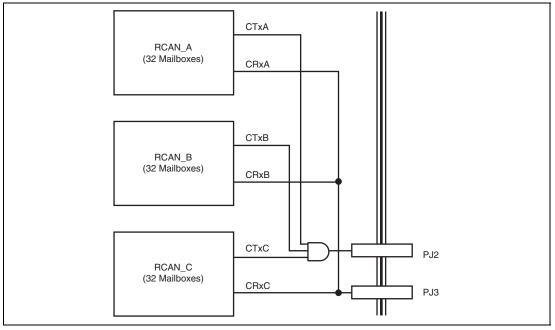


Figure 18.31 Connection Example when Using RCAN-TL1 as 1-Channel Module (96 Mailboxes × 1 Channel)

## 18.11 Usage Notes

#### 18.11.1 Notes on Port Setting for Multiple Channels Used as Single Channel

The RCAN-TL1 in this LSI has three channels and some of these channels can be used as a single channel. When using multiple channels as a single channel, keep the following in mind.

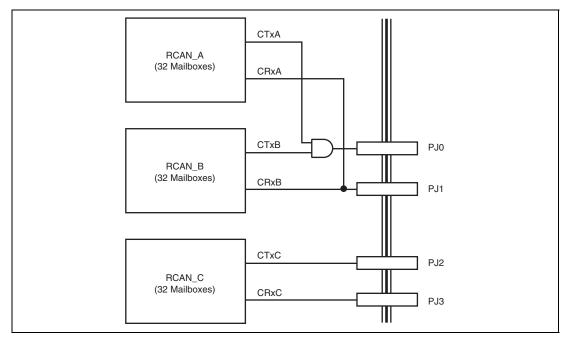


Figure 18.32 Connection Example when Using RCAN-TL1 as 2-Channel Module (64 Mailboxes × 1 Channel and 32 Mailboxes × 1 Channel)

No ACK error is detected even when any other nodes are not connected to the CAN bus. This
occurs when channel B transmits an ACK in the ACK field in response to a message channel
A has transmitted.

Channel B receives a message which channel A has transmitted on the CAN bus and then transmits an ACK in the ACK field. After that, channel A receives the ACK.

To avoid this, make channel B which is not currently used for transmission the listen-only mode (TST[2:0] = B'001) or the reset state (MCR0 = 1). With this setting, only a channel which transmits a message transmits an ACK.

- 2. Internal arbitration for channels A and B is independently controlled to determine the order of transmission.
  - Although the internal arbitration is performed on 31 Mailboxes at a time, it is not performed on 64 Mailboxes at a time even though multiple channels function as a single channel.
- Do not set the same transmission message ID in both channels A and B.
   Two messages may be transmitted from the two channels after arbitration on the CAN bus.

# Section 19 A/D Converter (ADC)

This LSI includes a 12-bit successive approximation A/D converter, which consists of two independent units (ADC\_A and ADC\_B). Up to 37 channel analog inputs can be selected by software.

#### 19.1 Features

- Resolution: 12 bits
- Input channels: 37 channels (ADC\_A: 28 channels (AN0 to AN27), ADC\_B: 9 channels (AN40 to AN48)
- Minimum conversion time

1.25  $\mu$ s/channel at P $\phi$  = 40MHz operation (conversion state = 50 states)

Low-speed setting: 2.5  $\mu$ s/channel (operation at P $\phi$  = 20 MHz, conversion state = 50 states)

High-speed setting: 1.25  $\mu$ s/channel (operation at P $\phi$  = 20 MHz, conversion state = 25 states)

• Two scan conversion modes are selectable

Single cycle scan mode: scanning only once

Continuous scan mode: scanning repeatedly

Channels to be scanned can be selected as desired, and A/D conversion is in ascending order of channel number (ADC A:  $ANO \rightarrow AN39$ , ADC B:  $AN40 \rightarrow AN48$ ).

A/D-converted value addition mode

The same channel is A/D converted continuously 2 to 4 times. The sum of the converted values is stored in the A/D data register. AN0 to AN7 and AN40 to AN47 can be used in A/D-converted value addition mode.

The use of the average of these results can improve the precision of A/D conversion, depending on the types of noise components that are present. This function, however, cannot guarantee an improvement in A/D conversion accuracy.

- Twenty-one 12-bit A/D data registers and sixteen 14-bit A/D data registers
- Sample and hold function

Each A/D converter module (ADC\_A and ADC\_B) includes a sample & hold circuit.

• Two types of scan conversion start

ADC\_A: Software trigger (ADST bit in ADCSR0), external trigger (ADTRG\_A) or ATU-III timer trigger (timer G4) can be selected.

ADC\_B: Software trigger (ADST bit in ADCSR1), external trigger (ADTRG\_B) or ATU-III timer trigger (timer G5) can be selected.

- Interrupt-triggered conversion
  - Independently from the scan conversion, it is possible to preferentially process channels requested by an ATU-III timer trigger or software trigger for A/D conversion. AN0 to AN15 and AN40 to AN47 support this function.
  - If an interrupt conversion and a scan conversion is requested simultaneously, the scan conversion is suspended and the A/D conversion is preferentially executed on the channel in which the interrupt conversion is requested. On completion of the interrupt conversion, the scan conversion is resumed from the A/D conversion on the interrupted channel.
- Supporting scan conversion end interrupt (ADI), interrupt conversion end interrupt (ADID), and DMA transfer function
  - On completion of scanning for scan conversion, a scan conversion end interrupt request (ADI) can be generated or the DMAC can be started. On completion of interrupt conversion on channels AN0 to AN15 or AN40 to AN47, an interrupt conversion end interrupt request (ADID0 to ADID15 and ADID40 to ADID47) can be generated, or A-DMAC (ADID0) or DMAC (ADID40 to ADID47) can be started up.
- Offset calibration function
  - The offset error due to temperature variation is corrected to achieve a high accuracy A/D conversion.
- Self-test function of A/D converter
  - The self-test function of A/D converter performs the A/D conversion of internally generated voltage values (AVref (AVrefh\_A, AVrefh\_B)  $\times$  0, AVref  $\times$  1/2, AVref  $\times$  1), and returns A/D converted values and information on converted voltages to A/D data registers Diag0 and Diag1. After that, whether or not the A/D converted values (the contents of the above A/D data registers) are in the normal range is checked by software to detect errors in the A/D converter.
- Programmable analog input voltage range
   Analog input voltage range is programmable via the AVrefh\_A pin.
- ADEND output
  - When channels AN0 and AN40 are used for scan conversion, the conversion timing signals are output via pins ADEND\_A and ADEND\_B.

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

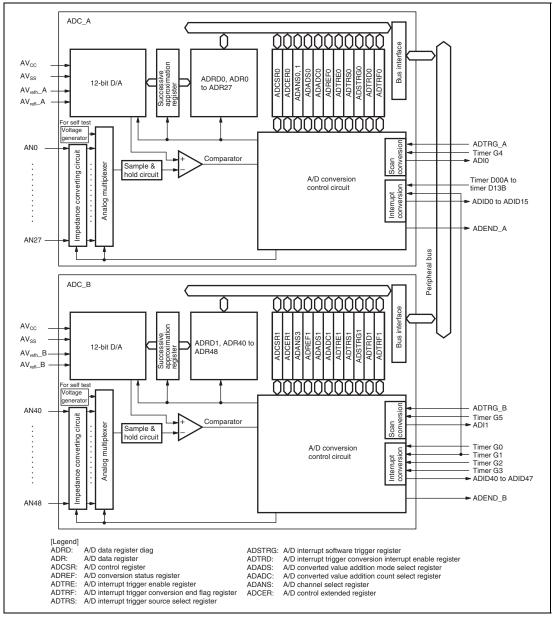


Figure 19.1 Block Diagram of A/D Converter

## 19.2 Input/Output Pins

Table 19.1 shows the pin configuration of the A/D converter.

For LSI reliability assurance, when using the A/D converter, ensure that the following is satisfied:

$$AV_{cc} = 5.0 V \pm 0.5 V$$
,  $AV_{ss} = V_{ss}$ 

When the A/D converter is not used, neither the  $AV_{cc}$  nor  $AV_{ss}$  pins must be left open. The voltage applied to the analog input pins must be within the following range.

$$AVrefl\_A \le ANn (n = 0 \text{ to } 27, 40 \text{ to } 48) \le AVrefh\_A$$

$$AVrefl\_B \le ANn (n = 0 \text{ to } 27, 40 \text{ to } 48) \le AVrefh\_B$$

**Table 19.1 Pin Configuration** 

| AV <sub>cc</sub> AV <sub>cc</sub> Input Analog power supply  AV <sub>ss</sub> AV <sub>ss</sub> Input Analog ground  AVREFL_A AVrefl_A Input Input pin for analog reference voltage for ADC (AVrefl_A < AVrefh_A)  AVREFH_A AVrefh_A Input Input pin for analog reference voltage for ADC (AVrefl_A < AVrefh_A)  AVREFL_B AVrefl_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AN_A0 AN0 Input Analog input channel 0  AN_A1 AN1 Input Analog input channel 1  AN_A2 AN2 Input Analog input channel 2 | _A<br>_B |
|--|----------|
| AVREFL_A AVrefl_A Input Input pin for analog reference voltage for ADC (AVrefl_A < AVrefh_A)  AVREFH_A AVrefh_A Input Input pin for analog reference voltage for ADC (AVrefl_A < AVrefh_A)  AVREFL_B AVrefl_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AVAGE AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AN_AO ANO Input Analog input channel 0  AN_A1 AN1 Input Analog input channel 1  | _A<br>_B |
| AVREFH_A AVrefh_A Input Input pin for analog reference voltage for ADC (AVrefl_A < AVrefh_A)  AVREFL_B AVrefl_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AN_AO ANO Input Analog input channel 0  AN_A1 AN1 Input Analog input channel 1   | _A<br>_B |
| AVREFH_A AVrefh_A Input Input pin for analog reference voltage for ADC (AVrefl_A < AVrefh_A)  AVREFL_B AVrefl_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AN_A0 AN0 Input Analog input channel 0  AN_A1 AN1 Input Analog input channel 1   | _B       |
| (AVrefl_A < AVrefh_A)  AVREFL_B AVrefl_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AN_A0 AN0 Input Analog input channel 0  AN_A1 AN1 Input Analog input channel 1  | _B       |
| AVREFL_B AVrefl_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AN_A0 AN0 Input Analog input channel 0  AN_A1 AN1 Input Analog input channel 1   |          |
| (AVrefl_B < AVrefh_B)  AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AN_A0 AN0 Input Analog input channel 0  AN_A1 AN1 Input Analog input channel 1  |          |
| AVREFH_B AVrefh_B Input Input pin for analog reference voltage for ADC (AVrefl_B < AVrefh_B)  AN_A0 AN0 Input Analog input channel 0  AN_A1 AN1 Input Analog input channel 1   | _B       |
| AN_A0 AN0 Input Analog input channel 0  AN_A1 AN1 Input Analog input channel 1   | _B       |
| AN_A0 AN0 Input Analog input channel 0 AN_A1 AN1 Input Analog input channel 1  |          |
| AN_A1 AN1 Input Analog input channel 1   |          |
|  |          |
| AN A2 AN2 Input Analog input channel 2   |          |
| Analog Input Charmer 2   |          |
| AN_A3 AN3 Input Analog input channel 3   |          |
| AN_A4 AN4 Input Analog input channel 4   |          |
| AN_A5 AN5 Input Analog input channel 5   |          |
| AN_A6 AN6 Input Analog input channel 6   |          |
| AN_A7 AN7 Input Analog input channel 7   |          |
| AN_A8 AN8 Input Analog input channel 8   |          |
| AN_A9 AN9 Input Analog input channel 9   |          |
| AN_A10 AN10 Input Analog input channel 10  |          |
| AN_A11 AN11 Input Analog input channel 11  |          |
| AN_A12 AN12 Input Analog input channel 12  |          |
| AN_A13 AN13 Input Analog input channel 13  |          |
| AN_A14 AN14 Input Analog input channel 14  |          |
| AN_A15 AN15 Input Analog input channel 15  |          |
| AN_A16 AN16 Input Analog input channel 16  |          |
| AN_A17 AN17 Input Analog input channel 17  |          |
| AN_A18 AN18 Input Analog input channel 18  |          |
| AN_A19 AN19 Input Analog input channel 19  |          |

| Pin Name | Symbol  | I/O    | Function  |
|----------|---------|--------|---|
| AN_A20   | AN20    | Input  | Analog input channel 20                                   |
| AN_A21   | AN21    | Input  | Analog input channel 21                                   |
| AN_A22   | AN22    | Input  | Analog input channel 22                                   |
| AN_A23   | AN23    | Input  | Analog input channel 23                                   |
| AN_A24   | AN24    | Input  | Analog input channel 24                                   |
| AN_A25   | AN25    | Input  | Analog input channel 25                                   |
| AN_A26   | AN26    | Input  | Analog input channel 26                                   |
| AN_A27   | AN27    | Input  | Analog input channel 27                                   |
| AN_B40   | AN40    | Input  | Analog input channel 40                                   |
| AN_B41   | AN41    | Input  | Analog input channel 41                                   |
| AN_B42   | AN42    | Input  | Analog input channel 42                                   |
| AN_B43   | AN43    | Input  | Analog input channel 43                                   |
| AN_B44   | AN44    | Input  | Analog input channel 44                                   |
| AN_B45   | AN45    | Input  | Analog input channel 45                                   |
| AN_B46   | AN46    | Input  | Analog input channel 46                                   |
| AN_B47   | AN47    | Input  | Analog input channel 47                                   |
| AN_B48   | AN48    | Input  | Analog input channel 48                                   |
| ADTRG_A  | ADTRG_A | Input  | Input pin for scan conversion trigger of ADC_A            |
| ADTRG_B  | ADTRG_B | Input  | Input pin for scan conversion trigger of ADC_B            |
| ADEND_A  | ADEND_A | Output | Output pin for monitoring AN0 conversion timing of ADC_A  |
| ADEND_B  | ADEND_B | Output | Output pin for monitoring AN40 conversion timing of ADC_B |

## 19.3 Functions Assigned to Each Channel

Table 19.2 lists the functions assigned to each channel.

Table 19.2 List of Functions Assigned to Each Channel

|  |                     | 1           |          |                              |         |   |                 |                  |                             |                                      |   | 1                                 |                                  |            |           |                                      | Interr                               | upi i | JUITVE                 | CISIOII                    |                          |                 |                      |
|--|---------------------|-------------|----------|------------------------------|---------|---|-----------------|------------------|-----------------------------|--------------------------------------|---|-----------------------------------|----------------------------------|------------|-----------|--------------------------------------|--------------------------------------|-------|------------------------|----------------------------|--------------------------|-----------------|----------------------|
| er<br>le/Channel                                     | A/D Conversio       | on .        | so<br>—  | Source 2 page                | er      | Self Test/Channel                             |                 | quest signal     | on (DMAC)                   | End flag (upon completion of a scan) | A/D data register<br>(Register storing A/D-converted value) | A/D converted value addition mode | liming monitor pin<br>ADEND_B)   | Source 1   | igger sou | Source 3                             | End interrupt DMA activation request |       | DIMA transfer function | Enable (channel selection) | Trigger source selection | End flag        | End interrupt enable |
| T#I#L  | Channel             |             | S        | S                            | So      | ő   | ĮĔ              | ¥                | ğ                           | Ē                                    | _ ₹   | a A                               | P                                | S          | တိ        |                                      | Ϋ́                                   | 1     | ₹                      | nat                        | igi                      | E E             | P P                  |
| Unit of A/D Converter<br>A/D Conversion Time/Channel |                     | Sym-<br>bol | Software | Pin                          | ATU-III | ADCER0 and<br>ADCER1/<br>ADANS0 and<br>ADANS1 | Scan conversion | End interrupt/DN | DMA transfer function (DMAC | End flag (upon c                     | A/D data register<br>(Register storing                      | ADADS0 and<br>ADADS1              | A/D conversion t<br>(ADEND_A and | ATU        | J-111     | Software<br>(ADSTRG0 and<br>ADSTRG1) | End interrupt DM                     | DMAC  | A-DMAC                 | ADTRE0 and ADTRE1          | ADTRS0 T<br>and ADTRS1   | ADTRF0 E ADTRF1 | ADTRD0 E             |
|  | Self test 0         | DIAG0       |          |                              |         | DIAGM<br>(ADCER0)                             |                 |                  |                             |                                      | ADRD0   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| L  | Analog input pin 0  | AN0         |          |                              |         | ANS0  |                 | Ш                |                             |                                      | ADR0  | ADS0                              | Yes                              | Timer D00A | Timer G1  | ADSTRG0                              | ADID0                                | No    | Yes                    | ADTRGE0                    | ADTRS0                   | ADTF0           | ADIDE0               |
| $I \mid L$   | Analog input pin 1  | AN1         |          |                              |         | ANS1  |                 |                  |                             | L                                    | ADR1  | ADS1                              | No                               | Timer D00B | No        | ADSTRG1                              | ADID1                                | No    | No                     | ADTRGE1                    | ADTRS1                   | ADTF1           | ADIDE1               |
| $  \cdot  $  | Analog input pin 2  | AN2         |          |                              |         | ANS2  |                 | Ш                |                             |                                      | ADR2  | ADS2                              | No                               | Timer D01A | No        | ADSTRG2                              | ADID2                                | No    | No                     | ADTRGE2                    | ADTRS2                   | ADTF2           | ADIDE2               |
| $  \cdot  $  | Analog input pin 3  | AN3         |          |                              |         | ANS3  |                 | Ш                |                             |                                      | ADR3  | ADS3                              | No                               | Timer D01B | No        | ADSTRG3                              | ADID3                                | No    | No                     | ADTRGE3                    | ADTRS3                   | ADTF3           | ADIDE3               |
| 1 I L  | Analog input pin 4  | AN4         |          |                              |         | ANS4  |                 | Ш                |                             |                                      | ADR4  | ADS4                              | No                               | Timer D02A | No        | ADSTRG4                              | ADID4                                | No    | No                     | ADTRGE4                    | ADTRS4                   | ADTF4           | ADIDE4               |
| 1 I L  | Analog input pin 5  | AN5         |          |                              |         | ANS5  |                 |                  |                             |                                      | ADR5  | ADS5                              | No                               | Timer D02B | No        | ADSTRG5                              | ADID5                                | No    | No                     | ADTRGE5                    | ADTRS5                   | ADTF5           | ADIDE5               |
| $I \mid L$   | Analog input pin 6  | AN6         |          |                              |         | ANS6  |                 |                  |                             | L                                    | ADR6  | ADS6                              | No                               | Timer D03A | No        | ADSTRG6                              | ADID6                                | No    | No                     | ADTRGE6                    | ADTRS6                   | ADTF6           | ADIDE6               |
| $I \mid L$   | Analog input pin 7  | AN7         |          |                              |         | ANS7  |                 |                  |                             | L                                    | ADR7  | ADS7                              | No                               | Timer D03B | No        | ADSTRG7                              | ADID7                                | No    | No                     | ADTRGE7                    | ADTRS7                   | ADTF7           | ADIDE7               |
| $I \mid L$   | Analog input pin 8  | AN8         |          |                              |         | ANS8  |                 |                  |                             | L                                    | ADR8  | No                                | No                               | Timer D10A | No        | ADSTRG8                              | ADID8                                | No    | No                     | ADTRGE8                    | ADTRS8                   | ADTF8           | ADIDE8               |
|  | Analog input pin 9  | AN9         |          |                              |         | ANS9  |                 |                  |                             |                                      | ADR9  | No                                | No                               | Timer D10B | No        | ADSTRG9                              | ADID9                                | No    | No                     | ADTRGE9                    | ADTRS9                   | ADTF9           | ADIDE9               |
|  | Analog input pin 10 | AN10        |          |                              |         | ANS10   |                 |                  |                             |                                      | ADR10   | No                                | No                               | Timer D11A | No        | ADSTRG10                             | ADID10                               | No    | No                     | ADTRGE10                   | ADTRS10                  | ADTF10          | ADIDE10              |
|  | Analog input pin 11 | AN11        |          |                              |         | ANS11   |                 |                  |                             | [                                    | ADR11   | No                                | No                               | Timer D11B | No        | ADSTRG11                             | ADID11                               | No    | No                     | ADTRGE11                   | ADTRS11                  | ADTF11          | ADIDE11              |
|  | Analog input pin 12 | AN12        |          |                              |         | ANS12   |                 |                  |                             | [                                    | ADR12   | No                                | No                               | Timer D12A | No        | ADSTRG12                             | ADID12                               | No    | No                     | ADTRGE12                   | ADTRS12                  | ADTF12          | ADIDE12              |
| ΙΙΓ  | Analog input pin 13 | AN13        |          |                              |         | ANS13   | 1               |                  |                             |                                      | ADR13   | No                                | No                               | Timer D12B | No        | ADSTRG13                             | ADID13                               | No    | No                     | ADTRGE13                   | ADTRS13                  | ADTF13          | ADIDE13              |
| ΙΙΓ  | Analog input pin 14 | AN14        |          |                              |         | ANS14   | 1               |                  |                             |                                      | ADR14   | No                                | No                               | Timer D13A | No        | ADSTRG14                             | ADID14                               | No    | No                     | ADTRGE14                   | ADTRS14                  | ADTF14          | ADIDE14              |
| ΙΙΓ  | Analog input pin 15 | AN15        |          |                              |         | ANS15   | 1               |                  |                             |                                      | ADR15   | No                                | No                               | Timer D13B | No        | ADSTRG15                             | ADID15                               | No    | No                     | ADTRGE15                   | ADTRS15                  | ADTF15          | ADIDE15              |
|  | Analog input pin 16 | AN16        |          |                              |         | ANS16   | scanning        | Ш                |                             |                                      | ADR16   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
|  | Analog input pin 17 | AN17        |          |                              |         | ANS17   | g               | Ш                |                             |                                      | ADR17   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| $I \mid \Gamma$                                      | Analog input pin 18 | AN18        |          |                              |         | ANS18   |                 | Ш                |                             |                                      | ADR18   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
|  | Analog input pin 19 | AN19        |          | 3                            |         | ANS19   | 1 🗟             | Ш                |                             | Ī                                    | ADR19   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
|  | Analog input pin 20 | AN20        |          | 2                            |         | ANS20   | or continuous   |                  |                             |                                      | ADR20   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| [3]  | Analog input pin 21 | AN21        |          | 힏                            |         | ANS21   | 18              |                  |                             | Ī                                    | ADR21   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| A/D converter_A (ADC_<br>2.5 µs/1.25 µs              | Analog input pin 22 | AN22        |          | External trigger_A (ADTRG_A) |         | ANS22   | ing             |                  |                             | 1                                    | ADR22   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| ×  | Analog input pin 23 | AN23        | SR0)     | ē                            |         | ANS23   | scanning        |                  |                             | _[                                   | ADR23   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| rter<br>5 µs   | Analog input pin 24 | AN24        | (ADC     | rig                          |         | ANS24   | e sc            |                  |                             | (ADREFO)                             | ADR24   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| /D convert<br>5 μs/1.25                              | Analog input pin 25 | AN25        | ₹        | alt                          | G4      | ANS25   | cycle           |                  |                             | <u>۳</u> [                           | ADR25   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| S 3  | Analog input pin 26 | AN26        | ADST     | terr                         | Timer   | ANS26   | Single          | 흳                | s                           | 1                                    | ADR26   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| 2.5  | Analog input pin 27 | AN27        | A        | ш                            | Ë       | ANS27   | iž              | ADIO             | Yes                         | ADF                                  | ADR27   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
|  | Self test 1         | DIAG1       | П        |                              | П       | DIAGM<br>(ADCER1)                             | Г               | П                | П                           |                                      | ADRD1   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |
| <u>a</u>   | Analog input pin 40 | AN40        |          |                              |         | ANS40   | L               |                  |                             |                                      | ADR40   | ADS40                             | Yes                              | Timer G0   | No        | ADSTRG40                             | ADID40                               | Yes   | No                     | ADTRGE40                   | ADTRS40                  | ADTF40          | ADIDE40              |
|  | Analog input pin 41 | AN41        |          | (B)                          |         | ANS41   | scan            |                  |                             | [                                    | ADR41   | ADS41                             | No                               | Timer G1   | No        | ADSTRG41                             | ADID41                               | Yes   | No                     | ADTRGE41                   | ADTRS41                  | ADTF41          | ADIDE41              |
| [S]  | Analog input pin 42 | AN42        |          | 띩                            |         | ANS42   | sno             |                  |                             | [                                    | ADR42   | ADS42                             | No                               | Timer G2   | No        | ADSTRG42                             | ADID42                               | Yes   | No                     | ADTRGE42                   | ADTRS42                  | ADTF42          | ADIDE42              |
| converter_B (ADC,                                    | Analog input pin 43 | AN43        |          | External trigger_B (ADTRG_   |         | ANS43   | continuous      |                  |                             | [                                    | ADR43   | ADS43                             | No                               | Timer G3   | No        | ADSTRG43                             | ADID43                               | Yes   | No                     | ADTRGE43                   | ADTRS43                  | ADTF43          | ADIDE43              |
| erte   | Analog input pin 44 | AN44        | اچ       | ၅                            |         | ANS44   |                 |                  |                             | _[                                   | ADR44   | ADS44                             | No                               | Timer G0   | No        | ADSTRG44                             | ADID44                               | Yes   | No                     | ADTRGE44                   | ADTRS44                  | ADTF44          | ADIDE44              |
| D conver<br>5 µs/1.25                                | Analog input pin 45 | AN45        | (ADCSR1) | gge                          |         | ANS45   | 15              |                  |                             | (ADREF1)                             | ADR45   | ADS45                             | No                               | Timer G1   | No        | ADSTRG45                             | ADID45                               | Yes   | No                     | ADTRGE45                   | ADTRS45                  | ADTF45          | ADIDE45              |
|  | Analog input pin 46 | AN46        | ğ        | E I                          | G5      | ANS46   | scan            |                  |                             | H [                                  | ADR46   | ADS46                             | No                               | Timer G2   | No        | ADSTRG46                             | ADID46                               | Yes   | No                     | ADTRGE46                   | ADTRS46                  | ADTF46          | ADIDE46              |
|  | Analog input pin 47 |             | 딞        | ern                          | Fimer ( | ANS47   | Single :        | <u> -</u>        | ,,                          |                                      | ADR47   | ADS47                             | No                               | Timer G3   | No        | ADSTRG47                             | ADID47                               | Yes   | No                     | ADTRGE47                   | ADTRS47                  | ADTF47          | ADIDE47              |
| Ш  | Analog input pin 48 | AN48        | ADST     | ŭ                            | Ë       | ANS48   | ŝ               | AD11             | Yes                         | ΑPF                                  | ADR48   | No                                | No                               | No         | No        | No                                   | No                                   | No    | No                     | No                         | No                       | No              | No                   |

# 19.4 Register Descriptions

The A/D converter has the following registers.

**Table 19.3 Register Configuration** 

| Register Name           | Symbol | R/W | Initial<br>Value | Address    | Unit  | Access<br>Size*1 |
|-------------------------|--------|-----|------------------|------------|-------|------------------|
| A/D data register Diag0 | ADRD0  | R   | H'0000           | H'FFFFE83E | ADC_A | 16               |
| A/D data register 0     | ADR0   | R   | H'0000           | H'FFFFE840 | ADC_A | 16               |
| A/D data register 1     | ADR1   | R   | H'0000           | H'FFFFE842 | ADC_A | 16               |
| A/D data register 2     | ADR2   | R   | H'0000           | H'FFFFE844 | ADC_A | 16               |
| A/D data register 3     | ADR3   | R   | H'0000           | H'FFFFE846 | ADC_A | 16               |
| A/D data register 4     | ADR4   | R   | H'0000           | H'FFFFE848 | ADC_A | 16               |
| A/D data register 5     | ADR5   | R   | H'0000           | H'FFFFE84A | ADC_A | 16               |
| A/D data register 6     | ADR6   | R   | H'0000           | H'FFFFE84C | ADC_A | 16               |
| A/D data register 7     | ADR7   | R   | H'0000           | H'FFFFE84E | ADC_A | 16               |
| A/D data register 8     | ADR8   | R   | H'0000           | H'FFFFE850 | ADC_A | 16               |
| A/D data register 9     | ADR9   | R   | H'0000           | H'FFFFE852 | ADC_A | 16               |
| A/D data register 10    | ADR10  | R   | H'0000           | H'FFFFE854 | ADC_A | 16               |
| A/D data register 11    | ADR11  | R   | H'0000           | H'FFFFE856 | ADC_A | 16               |
| A/D data register 12    | ADR12  | R   | H'0000           | H'FFFFE858 | ADC_A | 16               |
| A/D data register 13    | ADR13  | R   | H'0000           | H'FFFFE85A | ADC_A | 16               |
| A/D data register 14    | ADR14  | R   | H'0000           | H'FFFFE85C | ADC_A | 16               |
| A/D data register 15    | ADR15  | R   | H'0000           | H'FFFFE85E | ADC_A | 16               |
| A/D data register 16    | ADR16  | R   | H'0000           | H'FFFFE860 | ADC_A | 16               |
| A/D data register 17    | ADR17  | R   | H'0000           | H'FFFFE862 | ADC_A | 16               |
| A/D data register 18    | ADR18  | R   | H'0000           | H'FFFFE864 | ADC_A | 16               |
| A/D data register 19    | ADR19  | R   | H'0000           | H'FFFFE866 | ADC_A | 16               |
| A/D data register 20    | ADR20  | R   | H'0000           | H'FFFFE868 | ADC_A | 16               |
| A/D data register 21    | ADR21  | R   | H'0000           | H'FFFFE86A | ADC_A | 16               |
| A/D data register 22    | ADR22  | R   | H'0000           | H'FFFFE86C | ADC_A | 16               |
| A/D data register 23    | ADR23  | R   | H'0000           | H'FFFFE86E | ADC_A | 16               |
| A/D data register 24    | ADR24  | R   | H'0000           | H'FFFFE870 | ADC_A | 16               |

| Register Name  | Symbol  | R/W                 | Initial<br>Value | Address    | Unit  | Access<br>Size*1 |
|--|---------|---------------------|------------------|------------|-------|------------------|
| A/D data register 25   | ADR25   | R                   | H'0000           | H'FFFFE872 | ADC_A | 16               |
| A/D data register 26   | ADR26   | R                   | H'0000           | H'FFFFE874 | ADC_A | 16               |
| A/D data register 27   | ADR27   | R                   | H'0000           | H'FFFFE876 | ADC_A | 16               |
| A/D data register Diag1  | ADRD1   | R                   | H'0000           | H'FFFFEC3E | ADC_B | 16               |
| A/D data register 40   | ADR40   | R                   | H'0000           | H'FFFFEC40 | ADC_B | 16               |
| A/D data register 41   | ADR41   | R                   | H'0000           | H'FFFFEC42 | ADC_B | 16               |
| A/D data register 42   | ADR42   | R                   | H'0000           | H'FFFFEC44 | ADC_B | 16               |
| A/D data register 43   | ADR43   | R                   | H'0000           | H'FFFFEC46 | ADC_B | 16               |
| A/D data register 44   | ADR44   | R                   | H'0000           | H'FFFFEC48 | ADC_B | 16               |
| A/D data register 45   | ADR45   | R                   | H'0000           | H'FFFFEC4A | ADC_B | 16               |
| A/D data register 46   | ADR46   | R                   | H'0000           | H'FFFFEC4C | ADC_B | 16               |
| A/D data register 47   | ADR47   | R                   | H'0000           | H'FFFFEC4E | ADC_B | 16               |
| A/D data register 48   | ADR48   | R                   | H'0000           | H'FFFFEC50 | ADC_B | 16               |
| A/D control register 0   | ADCSR0  | R/W                 | H'00             | H'FFFFE800 | ADC_A | 8                |
| A/D control register 1   | ADCSR1  | R/W                 | H'00             | H'FFFFEC00 | ADC_B | 8                |
| A/D conversion status register 0                                 | ADREF0  | R/(W)*2             | H'00             | H'FFFFE802 | ADC_A | 8                |
| A/D conversion status register 1                                 | ADREF1  | R/(W)*2             | H'00             | H'FFFFEC02 | ADC_B | 8                |
| A/D interrupt trigger enable register 0                          | ADTRE0  | R/W                 | H'0000           | H'FFFFE804 | ADC_A | 8, 16            |
| A/D interrupt trigger enable register 1                          | ADTRE1  | R/W                 | H'00             | H'FFFFEC10 | ADC_B | 8                |
| A/D interrupt trigger conversion end flag register 0             | ADTRF0  | R/(W)*2             | H'0000           | H'FFFFE806 | ADC_A | 8, 16            |
| A/D interrupt trigger conversion end flag register 1             | ADTRF1  | R/(W)* <sup>2</sup> | H'00             | H'FFFFEC12 | ADC_B | 8                |
| A/D interrupt trigger source select register 0                   | ADTRS0  | R/W                 | H'0000           | H'FFFFE808 | ADC_A | 8, 16            |
| A/D interrupt trigger source select register 1                   | ADTRS1  | R/W                 | H'00             | H'FFFFEC14 | ADC_B | 8                |
| A/D interrupt software trigger register 0                        | ADSTRG0 | W                   | H'0000           | H'FFFFE80A | ADC_A | 8, 16            |
| A/D interrupt software trigger register 1                        | ADSTRG1 | W                   | H'00             | H'FFFFEC16 | ADC_B | 8                |
| A/D interrupt trigger conversion end interrupt enable register 0 | ADTRD0  | R/W                 | H'0000           | H'FFFFE80C | ADC_A | 8, 16            |

| Register Name  | Symbol | R/W | Initial<br>Value | Address    | Unit  | Access<br>Size*1 |
|--|--------|-----|------------------|------------|-------|------------------|
| A/D interrupt trigger conversion end interrupt enable register 1 | ADTRD1 | R/W | H'00             | H'FFFFEC18 | ADC_B | 8                |
| A/D-converted value addition mode select register 0              | ADADS0 | R/W | H'00             | H'FFFFE81C | ADC_A | 8                |
| A/D-converted value addition mode select register 1              | ADADS1 | R/W | H'00             | H'FFFFEC1C | ADC_B | 8                |
| A/D-converted value addition count selection register 0          | ADADC0 | R/W | H'00             | H'FFFFE81E | ADC_A | 8                |
| A/D-converted value addition count selection register 1          | ADADC1 | R/W | H'00             | H'FFFFEC1E | ADC_B | 8                |
| A/D channel select register 0                                    | ADANS0 | R/W | H'0000           | H'FFFFE820 | ADC_A | 8, 16,<br>32     |
| A/D channel select register 1                                    | ADANS1 | R/W | H'0000           | H'FFFFE822 | ADC_A | 8, 16,<br>32     |
| A/D channel select register 3                                    | ADANS3 | R/W | H'0000           | H'FFFFEC20 | ADC_B | 8, 16            |
| A/D control extended register 0                                  | ADCER0 | R/W | H'0000           | H'FFFFE830 | ADC_A | 8, 16            |
| A/D control extended register 1                                  | ADCER1 | R/W | H'0000           | H'FFFFEC30 | ADC_B | 8, 16            |

Notes: 1. 16-bit access can be made only on word boundaries, while 32-bit access can be made only on longword boundaries.

2. Only 0 can be written to clear the flag.

# 19.4.1 A/D Data Registers 0 to 27, 40 to 48, Diag0, and Diag1 (ADR0 to ADR27, ADR40 to ADR48, ADRD0, and ADRD1)

ADR0 to ADR27 and ADR40 to ADR48 are 16-bit read-only registers which store the A/D converted results of channels AN0 to AN27 and AN40 to AN48. ADRD0 and ADRD1 are 16-bit read-only registers which store the A/D converted results of ADC\_A and ADC\_B for self-test. ADR0 to ADR27, ADR40 to ADR48, ADRD0, and ADRD1 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

ADR0 to ADR27, ADR40 to ADR48, ADRD0 and ADRD1 use different formats depending on the setting values of the A/D data register format selection (ADRFMT) bit and A/D-converted value addition mode select bits (ADS0 to ADS7 and ADS40 to ADS47) Note that A/D-converted value addition mode can be written only in ADR0 to ADR7 and ADR40 to ADR47. ADRD0 and ADRD1 also include the self-test status bits.

#### (1) ADR0 to ADR7 and ADR40 to ADR47

The ADRFMT bit can be used to set either right- or left-alignment. For this operation, the AD11 to AD0 bits indicate a 12-bit A/D converted value. Other bits are reserved. These reserved bits are always read as 0. The write value should always be 0.

In A/D-converted value addition mode, any setting in the ADRFMT bit becomes invalid. In this case, the AD13 to AD0 bits indicate a value which is obtained by adding all converted values in A/D-converted value addition mode. Other bits are reserved. These reserved bits are always read as 0. The write value should always be 0.

The following minimum and maximum values apply to channels on which A/D-converted value addition mode is selected:

- First time:  $H'0000 \le ADRn (n = 0 \text{ to } 7, 40 \text{ to } 47) \le H'3FFC$
- Second time:  $H'0000 \le ADRn (n = 0 \text{ to } 7, 40 \text{ to } 47) \le H'7FF8$
- Third time:  $H'0000 \le ADRn (n = 0 \text{ to } 7, 40 \text{ to } 47) \le H'BFF4$
- Fourth time:  $H'0000 \le ADRn$  (n = 0 to 7, 40 to 47)  $\le H'FFF0$

## • When left-alignment is selected

| Bit:           | 15   | 14   | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3 | 2 | 1 | 0 |
|----------------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|
|                | AD11 | AD10 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 | - | - | - | - |
| Initial value: | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 |
| R/W:           | R    | R    | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R | R | R | R |

## • When right-alignment is selected

| Bit:           | 15 | 14 | 13 | 12 | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|----------------|----|----|----|----|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                | -  | -  | -  | -  | AD11 | AD10 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 |
| Initial value: | 0  | 0  | 0  | 0  | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W·           | R  | R  | R  | R  | R    | R    | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   |

## • When A/D-converted value addition mode is selected

| Bit:           | 15   | 14   | 13   | 12   | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1 | 0 |
|----------------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|
|                | AD13 | AD12 | AD11 | AD10 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 | - | - |
| Initial value: | 0    | 0    | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 |
| R/W:           | R    | R    | R    | R    | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R | R |

#### **(2)** ADR8 to ADR27 and ADR48

The ADRFMT bit can be used to set either right- or left-alignment. For this operation, the AD11 to AD0 bits indicate a 12-bit A/D converted value. Other bits are reserved. These reserved bits are always read as 0. The write value should always be 0.

ADR8 to ADR27 and ADR48 cannot be specified in A/D-converted value addition mode.

## When left-alignment is specified

| Bit:           | 15   | 14   | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3 | 2 | 1 | 0 |
|----------------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|
|                | AD11 | AD10 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 | - | - | - | - |
| Initial value: | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 |
| R/W:           | R    | R    | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R | R | R | R |

#### When right-alignment is specified

| Bit:                   | 15     | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|                        | 1      | -      | ı      | -      | AD11   | AD10   | AD9    | AD8    | AD7    | AD6    | AD5    | AD4    | AD3    | AD2    | AD1    | AD0    |
| Initial value:<br>R/W: | 0<br>R |

#### (3) ADRD0 and ADRD1

The ADRFMT bit can be used to specify either right or left-alignment. For this operation, bits AD11 to AD0 indicate a 12-bit A/D converted value. ADRD0 and ADRD1 include self-test status (DIAGST) bits. Other bits are reserved. These reserved bits are always read as 0. The write value should always be 0. ADR0 and ADR1 cannot be used in A/D-converted value addition mode.

## • When left-alignment is specified

| Bit:           | 15   | 14   | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3 | 2 | 1     | 0       |
|----------------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|-------|---------|
|                | AD11 | AD10 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 | - | - | DIAGS | ST[1:0] |
| Initial value: | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0     | 0       |
| R/W:           | R    | R    | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   | R | R | R     | R       |

| Bit     | Bit Name    | Initial<br>Value | R/W | Description   |
|---------|-------------|------------------|-----|---|
| 15 to 4 | AD11 to AD0 | All 0            | R   | 12-bit A/D-Converted Value  |
| 3, 2    | _           | All 0            | R   | Reserved  |
|         |             |                  |     | These bits are always read as 0. The write value should always be 0   |
| 1, 0    | DIAGST[1:0] | 00               | R   | Self-Test Status  |
|         |             |                  |     | These bits indicate the conversion voltage at which a self-test is performed. For details on self-test, see section 19.4.3, A/D Control Extended Registers 0 and 1 (ADCER0 and ADCER1). |
|         |             |                  |     | 00: No self-test has ever been performed since power-on.  |
|         |             |                  |     | 01: A self-test has been performed at a voltage AVref $\times$ 0.   |
|         |             |                  |     | 10: A self-test has been performed at a voltage AVref $\times$ 1/2.   |
|         |             |                  |     | 11: A self-test has been performed at a voltage AVref $\times$ 1.   |

# When right-alignment is specified

| Bit:           | 15    | 14      | 13 | 12 | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|----------------|-------|---------|----|----|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                | DIAGS | ST[1:0] | -  | -  | AD11 | AD10 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 |
| Initial value: | 0     | 0       | 0  | 0  | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| R/W:           | R     | R       | R  | R  | R    | R    | R   | R   | R   | R   | R   | R   | R   | R   | R   | R   |

| Bit     | Bit Name    | Initial<br>Value | R/W | Description   |
|---------|-------------|------------------|-----|---|
| 15, 14  | DIAGST[1:0] | 00               | R   | Self-test Status  |
|         |             |                  |     | These bits indicate the conversion voltage at which a self-test is performed. For details on self-test, see section 19.4.3, A/D Control Extended Registers 0 and 1 (ADCER0 and ADCER1). |
|         |             |                  |     | 00: No self-test has ever been performed since power-on.  |
|         |             |                  |     | 01: A self-test has been performed at a voltage AVref $\times$ 0.   |
|         |             |                  |     | 10: A self-test has been performed at a voltage AVref $\times$ 1/2.   |
|         |             |                  |     | 11: A self-test has been performed at a voltage AVref $\times$ 1.   |
| 13, 12  | _           | All 0            | R   | Reserved  |
|         |             |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 11 to 0 | AD11 to AD0 | All 0            | R   | 12-bit A/D-Converted Value  |

## 19.4.2 A/D Control Registers 0 and 1 (ADCSR0 and ADCSR1)

ADCSR0 and ADCSR1 are 8-bit readable/writable registers used to set scan conversion mode. ADCSR0 and ADCSR1 are initialized to H'00 by a power-on reset or in hardware standby mode.

| Bit:           | 7    | 6    | 5 | 4    | 3 | 2 | 1    | 0     |
|----------------|------|------|---|------|---|---|------|-------|
|                | ADST | ADCS | - | ADIE | - | - | TRGE | EXTRG |
| Initial value: | 0    | 0    | 0 | 0    | 0 | 0 | 0    | 0     |
| R/W:           | R/W  | R/W  | R | R/W  | R | R | R/W  | R/W   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 7   | ADST     | 0                | R/W | Scan Conversion Start  |
|     |          |                  |     | Starts or stops scan conversion.   |
|     |          |                  |     | When the ADST bit is set from 0 to 1, the A/D converter detects an ADST rising edge and then starts a scan conversion process. When the ADST bit is cleared from 1 to 0, the A/D converter detects an ADST falling edge and then stops a scan conversion process. The ADST bit does not affect interrupt conversion. To check whether a scan conversion is being performed, read the ADSCACT bit in ADREF.   |
|     |          |                  |     | 0: Stops a scan conversion process.  |
|     |          |                  |     | 1: Starts a scan conversion process.   |
| 6   | ADCS     | 0                | R/W | Scan Conversion Mode Select  |
|     |          |                  |     | Selects scan conversion mode. To prevent incorrect operation, the ADSCACT bit in ADREF must be cleared to 0 while the ADCS value is changed. Single-cycle scan mode performs scanning once and, upon its completion, stops the scan conversion process. Continuous scan mode repeats the scanning process indefinitely. The scan conversion can be stopped by writing 0 to the ADCS bit when the bit is set to 1. Channels are A/D-converted with the lowest channel first (ADC_A: AN0 → AN27, ADC_B: AN40 → AN47). In continuous scan mode, the conversion process returns to the first channel when all the selected channels have been converted. |
|     |          |                  |     | 0: Single-cycle scan mode  |
|     |          |                  |     | 1: Continuous scan mode  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 5    | _        | 0                | R   | Reserved   |
|      |          |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 4    | ADIE     | 0                | R/W | Interrupt Enable   |
|      |          |                  |     | Enables or disables generation of the A/D scan conversion end interrupt (ADI). To prevent incorrect operation, the ADSCACT bit in ADREF must be cleared to 0 while the ADIE value is changed.  |
|      |          |                  |     | When the ADF bit in ADREF is set to 1 upon completion of each scan in the scan conversion process, an ADI interrupt is generated if the ADIE bit is set to 1. The ADI interrupt can be cleared by clearing the ADF bit to 0 or clearing the ADIE bit to 0. |
|      |          |                  |     | <ol> <li>Disables ADI interrupt generation upon scanning completion.</li> </ol>  |
|      |          |                  |     | <ol> <li>Enables ADI interrupt generation upon scanning completion.</li> </ol>   |
| 3, 2 | _        | All 0            | R   | Reserved   |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 1    | TRGE     | 0                | R/W | Trigger Enable   |
|      |          |                  |     | Enables or disables scan conversion to be started by an external trigger (ADTGR_A, ADTGR_B) or ATU-III timer trigger (ADC_A: timer G4, ADC_B: timer G5).   |
|      |          |                  |     | <ol> <li>Disables scan conversion to be started by an<br/>external trigger or ATU-III timer trigger.</li> </ol>  |
|      |          |                  |     | <ol> <li>Enables scan conversion to be started by an<br/>external trigger or ATU-III timer trigger.</li> </ol>   |
| 0    | EXTRG    | 0                | R/W | Trigger Select   |
|      |          |                  |     | Selects a trigger source for scan conversion. Either the external trigger (ADTGR_A, ADTGR_B) or the ATU-III timer trigger (ADC_A: timer G4, ADC_B: timer G5) is selected.  |
|      |          |                  |     | <ol><li>Scan conversion is started by an ATU-III timer<br/>trigger.</li></ol>  |
|      |          |                  |     | 1: Scan conversion is started by an external trigger.  |

- Notes: 1. Starting ADC\_A and ADC\_B scan conversion simultaneously

  ADC\_A and ADC\_B scan conversion can be started simultaneously by writing 1 to the
  - TRGE bit and 0 to the EXTRG bit in both ADCSR0 and ADCSR1, and by inputting the timer G4 trigger and the timer G5 trigger from the ATU-III simultaneously. For details on the timer G settings, see section 13, Advanced Timer Unit III (ATU-III).
  - 2. Starting ADC\_A and ADC\_B scan conversion with different timing from each other ADC\_A and ADC\_B scan conversion can be started with different timing from each other by writing 1 to the TRGE bit and 0 to the EXTRG bit in ADCSR0 and ADCSR1, and by inputting the timer G4 trigger and the timer G5 trigger from the ATU-III with different timing from each other. For details on the timer G settings, see section 13, Advanced Timer Unit III (ATU-III).
  - 3. Staring an interrupt conversion and a scan conversion simultaneously If 1 is written to the TRGE bit and 0 is written to the EXTRG bit in ADCSR1, and 1 is written to the ADTRGE40 bit and the ADTRGE44 bit in the A/D interrupt trigger enable register (ADTRE1), and then the timer G5 trigger and the timer G0 trigger are simultaneously input from the ATU-III, the ADC\_B executes the following operations in the indicated sequence: AN40 interrupt conversion → AN44 interrupt conversion → scan conversion. To execute a scan conversion only, clear both the ADTRGE40 and ADTRGE44 bits to 0. Either AN40 or AN44 can also execute a single-channel interrupt conversion. Similar operations can also be accomplished through combinations of an ADC\_A scan conversion and an AN0 interrupt conversion by using the timer G4 trigger and the timer G1 trigger from the ATU-III.
  - 4. Starting a scan conversion using an external trigger If 1 is written to both the TRGE and EXTRG bits when high level signals are input to the external trigger pins (ADTRG\_A and ADTRG\_B), and then if a low-level pulse is input to either ADTRG\_A or ADTRG\_B, either ADC\_A or ADC\_B detects a pulse falling edge and starts the scan conversion process. In this case, the low pulse width must be 1.5Pφ clock or more.
  - Independent of the ADST bit, external triggers and ATU-III timer triggers, startup of a scan conversion is enabled when the ADSCACT bit in the A/D conversion status register (ADREF) is cleared to 0. The startup source for a scan conversion is not retained.
  - 6. Note on the cycle time of starting scan conversion or interrupt conversion by the timer trigger from the ATU-III: Set the ATU-III timer trigger cycle such that it is longer than the scan conversion time and the interrupt conversion time. (Example for scan conversion: in the case of conversion on a single channel, 56 states when bit CKS = 0 and 28 states when CKS = 1. Example for interrupt conversion: in the case of starting conversion on a single channel using one trigger source, 50 states when bit CKS = 0 and 25 states when CKS = 1.) For details on the timer trigger cycle setting, see section 13, Advanced Timer Unit III (ATU-III).

# 19.4.3 A/D Control Extended Registers 0 and 1 (ADCER0 and ADCER1)

ADCER0 and ADCER1 are 16-bit readable/writable registers to make settings for such functions as self-test mode. ADCER0 and ADCER1 are initialized to H'0000 by a power-on reset or in hardware standby mode.

| Bit:           | 15         | 14 | 13 | 12 | 11    | 10     | 9     | 8       | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0           |
|----------------|------------|----|----|----|-------|--------|-------|---------|-----|---|---|---|---|---|---|-------------|
|                | ADR<br>FMT | -  | -  | -  | DIAGM | DIAGLD | DIAGV | AL[1:0] | CKS | - | - | - | - | - | - | ITT<br>RGS* |
| Initial value: | 0          | 0  | 0  | 0  | 0     | 0      | 0     | 0       | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0           |
| R/W:           | R/W        | R  | R  | R  | R/W   | R/W    | R/W   | R/W     | R/W | R | R | R | R | R | R | R/W         |

Note: \* This bit is provided only in ADCER0 (ADC\_A).
Bit 0 in ADCER1 (ADC\_B) is reserved.

|          |          | Initial |     |   |
|----------|----------|---------|-----|---|
| Bit      | Bit Name | Value   | R/W | Description   |
| 15       | ADRFMT   | 0       | R/W | A/D Data Register Format Select   |
|          |          |         |     | The format of the A/D data register associated with a channel on which A/D-converted value addition mode is selected is fixed to left-alignment, irrespective of the ADRFMT bit value. For details on the format of the A/D data registers, see section 19.4.1, A/D Data Registers 0 to 27, 40 to 48, Diag0, and Diag1 (ADR0 to ADR27, ADR40 to ADR48, ADRD0, and ADRD1). |
|          |          |         |     | 0: Selects left-alignment.  |
|          |          |         |     | 1: Selects right-alignment.   |
| 14 to 12 | _        | All 0   | R   | Reserved  |
|          |          |         |     | These bits are always read as 0. The write value should always be 0.  |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 11  | DIAGM    | 0                | R/W | Self-Test Enable  |
|     |          |                  |     | Self-test function is used to detect errors in the A/D converter (ADC_A and ADC_B). The ADC_A and ADC_B convert the internally generated voltage values AVref (AVrefh_A and AVrefh_B) $\times$ 0, AVref $\times$ 1/2 and AVref $\times$ 1. Upon completion of the conversion, A/D converted values and information on converted voltages are stored in A/D data registers Diag0 and Diag1 (ADRD0 and ADRD1). After that, ADRD0 and ADRD1 are read by software to determine whether the A/D converted values are in the normal range or not. |
|     |          |                  |     | A self-test is performed before the lowest-numbered channel is converted in the scan conversion process. When the values of OFCM1 and OFCM0 bits are B'11 (calibration mode), self-test is not performed regardless of the DIAGM bit value. In each execution of a self-test, one of the three voltage values is converted, and the three voltage values are automatically rotated each time a self-test is executed. The execution time for self-test is equal to the A/D conversion time for a channel.                                   |
|     |          |                  |     | To prevent incorrect operation, the ADSCACT bit in ADREF must be cleared to 0 while the DIAGM value is changed.   |
|     |          |                  |     | 0: Does not perform a self-test for the A/D converter.  |
|     |          |                  |     | 1: Performs a self-test for the A/D converter.  |

|      |          | Initial |     |  |
|------|----------|---------|-----|--|
| Bit  | Bit Name | Value   | R/W | Description  |
| 10   | DIAGLD   | 0       | R/W | Self-test Mode Select  |
|      |          |         |     | Selects whether the three voltage values that are converted in the self-test process are to be rotated or to be fixed.   |
|      |          |         |     | Writing 0 to the DIAGLD bit causes a conversion to be performed with rotating voltage values in the following sequence: Avref $\times$ 0 $\rightarrow$ Avref $\times$ 1/2 $\rightarrow$ Avref $\times$ 1. If a self-test is performed beginning with AVref $\times$ 0 upon power-on reset, the voltage does not return to AVref $\times$ 0 even after the completion of the scan conversion. If a scan conversion is executed again, rotation is resumed from where it ended in the previous conversion operation. |
|      |          |         |     | Writing 1 to the DIAGLD bit causes a conversion at a fixed voltage value that is selected by the DIAGVAL1 and DIAGVAL0 bits in ADCER. (Automatic rotation is not performed.) Writing 0 again to the DIAGLD bit causes the start of rotation from a fixed voltage value (in a loading function).  |
|      |          |         |     | 0: Self-test is performed with automatic rotation.   |
|      |          |         |     | <ol> <li>Self-test is executed by using the fixed voltage<br/>value selected by DIAGVAL1 and DIAGVAL0.</li> </ol>  |
| 9, 8 | DIAGVAL  | 00      | R/W | Self-Test Voltage Select   |
|      | [1:0]    |         |     | For details, see the description of the DIAGLD bit. If<br>the values of these bits are B'00, executing self-test<br>by writing 1 to the DIAGLD is prohibited.  |
|      |          |         |     | 00: Reserved   |
|      |          |         |     | 01: Perform a self-test at a voltage AVref $\times$ 0  |
|      |          |         |     | 10: Perform a self-test at a voltage AVref $\times$ 1/2  |
|      |          |         |     | 11: Perform a self-test at a voltage AVref × 1   |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7      | CKS      | 0                | R/W | Clock Select   |
|        |          |                  |     | Selects A/D conversion time. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be 0 while changing the value of the CKS bit.   |
|        |          |                  |     | 0: A/D conversion time = 50 states (based on $P\phi$ )   |
|        |          |                  |     | 1: A/D conversion time = 25 states (based on $P\phi$ )   |
|        |          |                  |     | (Setting prohibited when two-times multiplication has been set for the peripheral clock)   |
| 6 to 1 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 0      | ITTRGS*  | 0                | R/W | Expanded Interrupt Conversion Trigger Source Selection   |
|        |          |                  |     | Selects the AN0 interrupt conversion trigger source between the timer D00A and timer G1 in the ATU-III. The ITTRGS bit becomes valid only when the ADTRGE0 bit in the ADTRE0 register is set to 1 and the ADTRS0 bit in the ADTRS0 register is cleared to 0. |
|        |          |                  |     | <ol><li>AN0 interrupt conversion is triggered by the timer<br/>D00A in the ATU-III.</li></ol>  |
|        |          |                  |     | 1: AN0 interrupt conversion is triggered by the timer G1 in the ATU-III.   |

Note: \* The ITTRGS bit (bit 0) is provided only in ADCER0 (ADC\_A). Bit 0 in ADCER1 is reserved.

#### 19.4.4 A/D Channel Select Registers 0, 1, and 3 (ADANS0, ADANS1, and ADANS3)

ADANS0, ADANS1, and ADANS3 are used to select channels that are subject to scan conversion. ADANS0, ADANS1, and ADANS3 are initialized to H'0000 by a power-on reset or in hardware standby mode. To prevent incorrect operation, the ADSCACT bit in ADREF must be cleared to 0 while the ADANS register values are changed.

Note: ADANS0, ADANS1, and ADANS3 are registers selects scan conversion channels; they are not used to select an interrupt conversion channel. An interrupt conversion channel is selected by the A/D interrupt trigger enable register (ADTRE).

If a channel is selected by both the ADANS and ADTRE registers, it is subject to conversion in both scan conversion and interrupt conversion. A channel that is selected only by the ADTRE register is excluded from the list of channels eligible for scan conversion, and only receives an interrupt conversion.

#### ADANS0

| Bit:           | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|----------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
|                | ANS15 | ANS14 | ANS13 | ANS12 | ANS11 | ANS10 | ANS9 | ANS8 | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 |
| Initial value: | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| R/M⋅           | R/M   | R/M   | R/M   | R/M   | R/M   | R/M   | R/M  | R/M  | R/W  | R/W  | R/M  | R/M  | R/M  | R/M  | R/M  | R/M  |

| Bit     | Bit Name         | Initial<br>Value | R/W | Description  |
|---------|------------------|------------------|-----|--|
| 15 to 0 | ANS15 to<br>ANS0 | All 0            | R/W | Setting the ANSn bit to 1 selects ANn. The correspondence between ANn and the ANSn bit is shown in table 19.2. |
|         |                  |                  |     | 0: ANn is not selected.  |
|         |                  |                  |     | 1: ANn is selected.  |

Apr 01, 2014

# ADANS1

| Bit:           | 15 | 14 | 13 | 12 | 11    | 10    | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|----|----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | -  | -  | 1  | -  | ANS27 | ANS26 | ANS25 | ANS24 | ANS23 | ANS22 | ANS21 | ANS20 | ANS19 | ANS18 | ANS17 | ANS16 |
| Initial value: | 0  | 0  | 0  | 0  | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R  | R  | R  | R  | R/W   |

| Bit      | Bit Name          | Initial<br>Value | R/W | Description  |
|----------|-------------------|------------------|-----|--|
| 15 to 12 | _                 | All 0            | R   | Reserved   |
|          |                   |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 11 to 0  | ANS27 to<br>ANS16 | All 0            | R/W | Setting the ANSn bit to 1 selects ANn. The correspondence between ANn and the ANSn bit is shown in table 19.2. |
|          |                   |                  |     | 0: ANn is not selected.  |
|          |                   |                  |     | 1: ANn is selected.  |

Note: n = 16 to 27

# ADANS3

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|----|----|----|----|---|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | -  | -  | -  | -  | -  | -  | - | ANS48 | ANS47 | ANS46 | ANS45 | ANS44 | ANS43 | ANS42 | ANS41 | ANS40 |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R/W   |

|         |                   | Initial |     |  |
|---------|-------------------|---------|-----|--|
| Bit     | Bit Name          | Value   | R/W | Description  |
| 15 to 9 | _                 | All 0   | R   | Reserved   |
|         |                   |         |     | These bits are always read as 0. The write value should always be 0.   |
| 8 to 0  | ANS48 to<br>ANS40 | All 0   | R/W | Setting the ANSn bit to 1 selects ANn. The correspondence between ANn and the ANSn bit is shown in table 19.2. |
|         |                   |         |     | 0: ANn is not selected.  |
|         |                   |         |     | 1: ANn is selected.  |

# 19.4.5 A/D Conversion Status Registers 0 and 1 (ADREF0 and ADREF1)

ADREF0 and ADREF1 indicate the status of the A/D converter. ADREF0 and ADREF1 are initialized to H'00 by a power-on reset or in hardware standby mode.

| Bit:           | 7           | 6           | 5 | 4 | 3 | 2 | 1 | 0      |
|----------------|-------------|-------------|---|---|---|---|---|--------|
|                | ADS<br>CACT | ADI<br>TACT | - | - | - | - | - | ADF    |
| Initial value: | 0           | 0           | 0 | 0 | 0 | 0 | 0 | 0      |
| R/W:           | R           | R           | R | R | R | R | R | R/(W)* |

Note: \* Only 0 can be written to clear the flag after reading the flag as 1.

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 7   | ADSCACT  | 0                | R   | Scan Conversion Status  |
|     |          |                  |     | Indicates whether the scan conversion process is in<br>the idle state or it is being executed. This is a read-<br>only bit and cannot be written.   |
|     |          |                  |     | If an interrupt conversion is started during a scan conversion, the A/D converter stops the scan conversion process and preferentially executes the interrupt conversion. However, until such time that all scan conversion is completed, the ADSCACT bit maintains to be set to 1 and is not cleared to 0. |
|     |          |                  |     | 0: Indicates that the Scan conversion process is in idle state.   |
|     |          |                  |     | 1: Indicates that the Scan conversion process is being executed.  |
| 6   | ADITACT  | 0                | R   | Interrupt Conversion Status   |
|     |          |                  |     | Indicates whether the interrupt conversion process is in the idle state or it is being executed. This is a read-only bit and cannot be written.   |
|     |          |                  |     | The ADSCACT and ADITACT bits can indicate the status of the ADC_A and ADC_B. For details, see table 19.4.   |
|     |          |                  |     | 0: Indicates that the interrupt conversion process is in idle state.  |
|     |          |                  |     | Indicates that the Interrupt conversion process is being executed.  |

| Bit    | Bit Name | Initial<br>Value | R/W    | Description  |
|--------|----------|------------------|--------|--|
| 5 to 1 | _        | All 0            | R      | Reserved   |
|        |          |                  |        | These bits are always read as 0. The write value should always be 0.   |
| 0      | ADF      | 0                | R/(W)* | Single Scan End Flag   |
|        |          |                  |        | This bit is set to 1 each time scanning ends in the scan conversion process (when all selected channels are converted). 1 cannot be written to this bit.   |
|        |          |                  |        | When the ADF bit is set to 1, either a scan conversion end interrupt or a DMA transfer request to the DMAC can be generated. In this manner, processing such as storing the contents of the A/D data register to the RAM can be implemented by means of either software or the DMAC. |
|        |          |                  |        | 0: Indicates that the scan conversion process is in idle state.  |
|        |          |                  |        | <ol> <li>Indicates that a single scan has been completed<br/>and the A/D-converted values on all selected ANn<br/>channels have been transferred to ADRn.</li> </ol>   |
|        |          |                  |        | [Clearing conditions]  |
|        |          |                  |        | • 0 is written to this bit after reading 1.  |
|        |          |                  |        | The DMAC is started-up by the ADI.   |
|        |          |                  |        | [Setting condition]  |
|        |          |                  |        | All analog conversion has been completed during  |
|        |          |                  |        | each scanning in scan conversion process.  |

Note: \* Only 0 can be written to clear the flag after reading the flag as 1.

Table 19.4 Relationship between ADC\_A and ADC\_B Status and ADSCACT and ADITACT

| ADSCACT | ADITACT | ADC_A and ADC_B Status | Source of Scan Conversion | Source of Interrupt Conversion |
|---------|---------|------------------------|---------------------------|--------------------------------|
| 0       | 0       | Idle state             | No                        | No                             |
|         | 1       | Interrupt conversion   | No                        | Yes                            |
| 1       | 0       | Scan conversion        | Yes                       | No                             |
|         | 1       | Interrupt conversion   | Yes                       | Yes                            |

# 19.4.6 A/D-Converted Value Addition Mode Select Registers 0 and 1 (ADADS0 and ADADS1)

ADADS0 and ADADS1 select ANn (n = 0 to 7, 40 to 47) on which A/D conversion is performed successively 2 to 4 times and then converted values are added (integrated). ADADS0 and ADADS1 are initialized to H'00 by a power-on reset or in hardware standby mode.

#### • ADADS0

| Bit:           | 7    | 6    | 5    | 4    | 4 3  |      | 1 0  |      |
|----------------|------|------|------|------|------|------|------|------|
|                | ADS7 | ADS6 | ADS5 | ADS4 | ADS3 | ADS2 | ADS1 | ADS0 |
| Initial value: | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| R/W·           | R/W  |

|        |          | R/W: R/V         |  | R/W R/W R/W R/W R/W   |  |  |  |  |  |
|--------|----------|------------------|--|---|--|--|--|--|--|
| Bit    | Bit Name | Initial<br>Value | R/W  | Description   |  |  |  |  |  |
| 7 to 0 | ADS7 to  | All 0            | R/W  | A/D-Converted Value Addition Channel Select   |  |  |  |  |  |
|        | ADS0     |                  |  | When the ADSn bit is set to 1, the A/D converter performs conversion on ANn successively 2 to 4 times and returns the added (integrated) conversion results to the A/D data register. If the ADSn bit is cleared to 0, the A/D converter performs a normal 1-time conversion of ANn and returns the conversion result to the A/D data register. Irrespective of whether a scan conversion or an interrupt conversion is to be performed, the A/D converter determines whether or not to perform an addition according to the ADSn value. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADSn bit value is changed. |  |  |  |  |  |
|        |          |                  |  | The correspondence between ANn and the ANSn bit is shown in table 19.2. How to select the addition count is described in section 19.4.7, A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1).   |  |  |  |  |  |
|        |          |                  |  | 0: A/D-converted value addition mode is not selected.   |  |  |  |  |  |
|        |          |                  | A/D-converted value addition mode (succeed) to 4 times of addition) is selected. |   |  |  |  |  |  |

Note: n = 0 to 7

# • ADADS1

| Bit:           | 7 6   |       | 5 4   |       | 3 2   |       | 1 0   |       |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | ADS47 | ADS46 | ADS45 | ADS44 | ADS43 | ADS42 | ADS41 | ADS40 |
| Initial value: | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R/W   |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7 to 0 | ADS47 to | All 0            | R/W | A/D-Converted Value Addition Channel Select   |
|        | ADS40    |                  |     | When the ADSn bit is set to 1, the A/D converter performs conversion on ANn successively 2 to 4 times and returns the added (integrated) conversion results to the A/D data register. If the ADSn bit is cleared to 0, the A/D converter performs a normal 1-time conversion of ANn and returns the conversion result to the A/D data register. Irrespective of whether a scan conversion or an interrupt conversion is to be performed, the A/D converter determines whether or not to perform an addition according to the ADSn value. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADSn bit value is changed. |
|        |          |                  |     | The correspondence between ANn and the ANSn bit is shown in table 19.2. How to select the addition count is described in section 19.4.7, A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1).   |
|        |          |                  |     | 0: A/D-converted value addition mode is not selected.   |
|        |          |                  |     | 1: A/D-converted value addition mode (successively 2 to 4 times of addition) is selected.   |

Figure 19.2 shows a scan conversion sequence in which both the ADS42 and ADS46 bits are set to 1, based on the assumption that the addition count is set to 4, and that channels AN40 to AN47 are selected. The conversion process begins with AN40. The AN42 conversion is performed successively 4 times, and the addition (integration) value is returned to the data register, after which the AN43 conversion process is started. If an interrupt conversion is requested in the midst of a scan conversion, the scan conversion process is stopped and an A/D conversion is preferentially executed on the channel in which an interrupt conversion was requested. Upon completion of the interrupt conversion, the scan conversion process is resumed from the A/D conversion on the interrupted channel. However, if the ADSn bit of the interrupted channel (ANn) is set to 1, even if addition has been performed at least once (two to four times), the conversion is restarted from the 1st conversion.

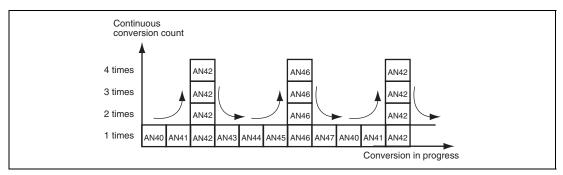


Figure 19.2 Scan Conversion Sequence with ADS42 and ADS46 Bits = 1

# 19.4.7 A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1)

ADADC0 and ADADC1 set the addition count for channels for which A/D-converted value addition mode is selected. ADADC0 and ADADC1 are initialized to H'00 by a power-on reset or in hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1   | 0     |
|----------------|---|---|---|---|---|---|-----|-------|
|                | - | - | - | - | - | - | ADC | [1:0] |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0     |
| R/W·           | R | R | R | R | R | R | R/W | R/W   |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 2 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 1, 0   | ADC[1:0] | 00               | R/W | Addition Count Select  |
|        |          |                  |     | These bits select the number of additions to be performed in A/D-converted value addition mode. These bits have no effect on the A/D conversion of channels for which A/D-converted value addition mode is not selected. |
|        |          |                  |     | To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADC1 and ADC0 bit values are changed.  |
|        |          |                  |     | 00: 1-time conversion (normal conversion)  |
|        |          |                  |     | 01: 2-time conversion  |
|        |          |                  |     | 10: 3-time conversion  |
|        |          |                  |     | 11: 4-time conversion  |

#### A/D Interrupt Trigger Enable Registers 0 and 1 (ADTRE0 and ADTRE1) 19.4.8

ADTRE0 and ADTRE1 enable or disable an interrupt conversion request for AN0 to AN15 and AN40 to AN47. Channels for which interrupt conversion is enabled are subjected to an interrupt conversion when a corresponding interrupt conversion request is input. ADTRE0 and ADADC1 are initialized to H'0000 and H'00, respectively, by a power-on reset or in hardware standby mode.

#### ADTRE0

| Bit:           | 15           | 14           | 13           | 12           | 11           | 10           | 9           | 8           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | ADTR<br>GE15 | ADTR<br>GE14 | ADTR<br>GE13 | ADTR<br>GE12 | ADTR<br>GE11 | ADTR<br>GE10 | ADTR<br>GE9 | ADTR<br>GE8 | ADTR<br>GE7 | ADTR<br>GE6 | ADTR<br>GE5 | ADTR<br>GE4 | ADTR<br>GE3 | ADTR<br>GE2 | ADTR<br>GE1 | ADTR<br>GE0 |
| Initial value: | 0            | 0            | 0            | 0            | 0            | 0            | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W          | R/W          | R/W          | R/W          | R/W          | R/W          | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         |

| Bit     | Bit Name   | Initial<br>Value | R/W | Description  |
|---------|------------|------------------|-----|--|
| 15 to 0 | ADTRGE15   | All 0            | R/W | Interrupt Conversion Request Enable  |
| t       | to ADTRGE0 |                  |     | Setting the ADTRGEn bit to 1 enables the interrupt conversion request to the corresponding ANn channel.        |
|         |            |                  |     | The correspondence among ANn, the ADTRGEn bit and the interrupt request trigger source is shown in table 19.2. |
|         |            |                  |     | 0: Disables an interrupt conversion request to ANn by ATU-III timer or software trigger (ADSTRGn).             |
|         |            |                  |     | 1: Enables an interrupt conversion request to ANn by ATU-III timer or software trigger (ADSTRGn).              |

n = 0 to 15Note:

# ADTRE1

| Bit:           | 7            | 6            | 5            | 4            | 3            | 2            | 1            | 0            |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                | ADTR<br>GE47 | ADTR<br>GE46 | ADTR<br>GE45 | ADTR<br>GE44 | ADTR<br>GE43 | ADTR<br>GE42 | ADTR<br>GE41 | ADTR<br>GE40 |
| Initial value: | 0            | 0            | 0            | 0            | 0            | 0            | 0            | 0            |
| R/W:           | R/W          |

| Bit    | Bit Name       | Initial<br>Value | R/W | Description   |
|--------|----------------|------------------|-----|---|
| 7 to 0 | ADTRGE47       | All 0            | R/W | Interrupt Conversion Request Enable   |
|        | to<br>ADTRGE40 |                  |     | Setting the ADTRGEn bit to 1 enables the interrupt conversion request to the corresponding ANn channel.   |
|        |                |                  |     | The correspondence among ANn, the ADTRGEn bit and the interrupt request trigger source is shown in table 19.2.  |
|        |                |                  |     | If the timer G0 signal is input when both the ADTRGE40 and ADTRGE44 bits in ADTRE1 are set to 1 and the timer G0 in the ATU-III is selected as an interrupt request source, the A/D converter converts AN44 after converting AN40. Similar operations can also be performed by the interrupt conversion of AN41 and AN45 on timer G1, of AN42 and AN46 on timer G2, and of AN43 and AN47 on timer G3. |
|        |                |                  |     | 0: Disables an interrupt conversion request to ANn by ATU-III timer or software trigger (ADSTRGn).  |
|        |                |                  |     | Enables an interrupt conversion request to ANn by ATU-III timer or software trigger (ADSTRGn).  |

# 19.4.9 A/D Interrupt Trigger Source Select Registers 0 and 1 (ADTRS0 and ADTRS1)

ADTRS0 and ADTRS1 select the trigger source for interrupt conversion. Either an ATU-III timer trigger or a software trigger caused by writing to the ADSTRG0 and ADSTRG1 registers can be selected. ADTRS0 and ADTRS1 are initialized to H'0000 and H'00 respectively by a power-on reset or in hardware standby mode.

#### ADTRS0

| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | ADT<br>RS15 | ADT<br>RS14 | ADT<br>RS13 | ADT<br>RS12 | ADT<br>RS11 | ADT<br>RS10 | ADT<br>RS9 | ADT<br>RS8 | ADT<br>RS7 | ADT<br>RS6 | ADT<br>RS5 | ADT<br>RS4 | ADT<br>RS3 | ADT<br>RS2 | ADT<br>RS1 | ADT<br>RS0 |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

|         |            | Initial |     |   |
|---------|------------|---------|-----|---|
| Bit     | Bit Name   | Value   | R/W | Description   |
| 15 to 1 | ADTRS15 to | All 0   | R/W | Interrupt Conversion Trigger Source Select  |
|         | ADTRS1     |         |     | If the ADTRSn bit is cleared to 0 and the ADTREn bit in the A/D interrupt trigger enable register is set to 1, the A/D converter detects an edge and begins the interrupt conversion on ANn when an ATU-III timer trigger is input. |
|         |            |         |     | If the ADTRSn bit is set to 1, the A/D converter detects an edge and begins the interrupt conversion on ANn when 1 is written to the ADSTRGn bit in the A/D interrupt software trigger register.                                    |
|         |            |         |     | The correspondence among the ADTRSn bit, ANn, and the interrupt request trigger source is shown in table 19.2.  |
|         |            |         |     | 0: Uses the timer D in the ATU-III as an ANn interrupt conversion request source.   |
|         |            |         |     | <ol> <li>Uses a software trigger (ADSTRGn) as an ANn<br/>interrupt conversion request source.</li> </ol>  |

| Bit | Bit Name   | Initial<br>Value  | R/W | Description   |
|-----|--|---|-----|---|
| 0   | ADTRS0   | 0   | R/W | Interrupt Conversion Trigger Source Select  |
|     | used as an ATU cleared to 0 and to 0, the timer I conversion requise cleared to 0 at timer G1 is selected. | For AN0, either the timer D00A or timer G1 can be used as an ATU-III timer trigger. If the ADTRS0 bit is cleared to 0 and the ITTRGS bit in ADCER0 is cleared to 0, the timer D00A is selected as an interrupt conversion request trigger source. If the ADTRS0 bit is cleared to 0 and the ITTRGS bit is set to 1, the timer G1 is selected as an interrupt conversion request trigger source. |     |   |
|     |  |   |     | <ol><li>Uses the timer D00A or timer G1 in the ATU-III as<br/>an AN0 interrupt conversion request source.</li></ol> |
|     |  |   |     | <ol> <li>Uses the software trigger (ADSTRGn) as an AN0<br/>interrupt conversion request source.</li> </ol>          |

# ADTRS1

| Bit:           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | ADT<br>RS47 | ADT<br>RS46 | ADT<br>RS45 | ADT<br>RS44 | ADT<br>RS43 | ADT<br>RS42 | ADT<br>RS41 | ADT<br>RS40 |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |

| Bit    | Bit Name   | Initial<br>Value | R/W | Description   |
|--------|------------|------------------|-----|---|
| 7 to 0 | ADTRS47 to |                  | R/W | Interrupt Conversion Trigger Source Select  |
| P      | ADTRS40    |                  |     | If the ADTRSn bit is cleared to 0 and the ADTREn bit in the A/D interrupt trigger enable register is set to 1, the A/D converter detects an edge and begins the interrupt conversion on ANn when an ATU-III timer trigger is input. |
|        |            |                  |     | If the ADTRSn bit is set to 1, the A/D converter detects an edge and begins the interrupt conversion on ANn when 1 is written to the ADSTRGn bit in the A/D interrupt software trigger register.                                    |
|        |            |                  |     | The correspondence among the ADTRSn bit, ANn, and the interrupt request trigger source is shown in table 19.2.  |
|        |            |                  |     | 0: Uses the timer G in ATU-III as an ANn interrupt conversion request source.   |
|        |            |                  |     | <ol> <li>Uses the software trigger (ADSTRGn) as an ANn<br/>interrupt conversion request source.</li> </ol>  |

# A/D Interrupt Software Trigger Registers 0 and 1 (ADSTRG0 and ADSTRG1)

ADSTRG0 and ADSTRG1 start an interrupt conversion by software. ADSTRG0 and ADSTRG1 are write-only registers and they are always read as 0s.

#### ADSTRG0

| Bit:           | 15           | 14 | 13           | 12 | 11           | 10           | 9 | 8           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|----------------|--------------|----|--------------|----|--------------|--------------|---|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | ADST<br>RG15 |    | ADST<br>RG13 |    | ADST<br>RG11 | ADST<br>RG10 |   | ADST<br>RG8 | ADST<br>RG7 | ADST<br>RG6 | ADST<br>RG5 | ADST<br>RG4 | ADST<br>RG3 | ADST<br>RG2 | ADST<br>RG1 | ADST<br>RG0 |
| Initial value: | 0            | 0  | 0            | 0  | 0            | 0            | 0 | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | W            | W  | W            | W  | W            | W            | W | W           | W           | W           | W           | W           | W           | W           | W           | W           |

Initial Value

| Bit     | Bit Name   | Value | R/W | Description  |
|---------|------------|-------|-----|--|
| 15 to 0 | ADSTRG15   | All 0 | W   | Interrupt Conversion Software Trigger  |
|         | to ADSTRGO |       |     | If 1 is written to the ADSTRGn bit when the ADTRSn bit in the A/D interrupt trigger source select register associated with the ANn is set to 1 and the ADTREn bit in the A/D interrupt trigger enable register is set to 1, the A/D converter detects an edge and starts the interrupt conversion of ANn. If an ANn is not subject to an interrupt conversion request, 0 should be written to it. ANn channels for which 0s are written are not affected by any of these operations. If an interrupt conversion is requested, the interrupt source is stored in the internal circuit in units of ANn. If an interrupt conversion on ANn with a source is performed and completed, the source associated with the ANn is cleared. Consequently, if 1 is written to the ADSTRGn and subsequently 0 is written to it, the source associated with the ANn is not cleared, and, therefore, the interrupt conversion is executed. It should be noted, however, that if a source is pending on ANn and then 1 is written to the ADSTRGn, it does not follow that an interrupt conversion is performed on the ANn twice. |

There is one source per channel. The same rule also applies to the execution of an interrupt conversion in response to a request from an ATU-III timer trigger. See table 19.2 for correspondence between ADSTRGn and ANn.

- 0: No interrupt conversion is request (software trigger) on the ANn.
- 1: Interrupt conversion is requested (software trigger) on the ANn.

# ADSTRG1

| Bit:           | 7            | -            | 5            |              |              | 2            |              | 0            |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                | ADST<br>RG47 | ADST<br>RG46 | ADST<br>RG45 | ADST<br>RG44 | ADST<br>RG43 | ADST<br>RG42 | ADST<br>RG41 | ADST<br>RG40 |
| Initial value: | 0            | 0            | 0            | 0            | 0            | 0            | 0            | 0            |
| R/W:           | W            | W            | W            | W            | W            | W            | W            | W            |

| Bit    | Bit Name                   | Initial<br>Value | R/W | Description  |
|--------|----------------------------|------------------|-----|--|
| 7 to 0 | ADSTRG47<br>to<br>ADSTRG40 | All 0            | W   | Interrupt Conversion Software Trigger  If 1 is written to the ADSTRGn bit when the ADTRSn bit in the A/D interrupt trigger source select register associated with the ANn is set to 1 and the ADTREn bit in the A/D interrupt trigger enable register is set to 1, the A/D converter detects an edge and starts the interrupt conversion of ANn. If an ANn is not subject to an interrupt conversion request, 0 should be written to it. ANn channels for which 0s are written are not affected by any of these operations. If an interrupt conversion is requested, the interrupt source is stored in the internal circuit in units of ANn. If an interrupt conversion on ANn with a source is performed and completed, the source associated with the ANn is cleared. Consequently, if 1 is written to the ADSTRGn and subsequently 0 is written to it, the source associated with the ANn is not cleared, and, therefore, the interrupt conversion is executed. It should be noted, however, that if a source is pending on ANn and then 1 is written to the ADSTRGn, it does not follow that an interrupt conversion is performed on the |
|        |                            |                  |     | <ul> <li>ANn twice.</li> <li>There is one source per channel. The same rule also applies to the execution of an interrupt conversion in response to a request from an ATU-III timer trigger.</li> <li>See table 19.2 for correspondence between ADSTRGn and ANn.</li> <li>O: No interrupt conversion is request ed(software trigger) on the ANn.</li> </ul>  |
|        |                            |                  |     | <ol> <li>Interrupt conversion is requested (software trigger)<br/>on the ANn.</li> </ol>   |

# 19.4.11 A/D Interrupt Trigger Conversion End Flag Registers 0 and 1 (ADTRF0 and ADTRF1)

ADTRF0 and ADTRF1 indicate that an interrupt conversion has been completed. When an interrupt conversion has been completed, the ADTFn bit corresponding to the channel (ANn) is set to 1. ADTRF0 and ADTRF1 are initialized to H'0000 and H'00 respectively by a power-on reset or in hardware standby mode.

#### ADTRF0

| Bit:   | 15         | 14         | 13         | 12         | 11         | 10         | 9         | 8         | 7         | 6         | 5         | 4         | 3         | 2         | 1         | 0         |
|--|------------|------------|------------|------------|------------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
|  | ADT<br>F15 | ADT<br>F14 | ADT<br>F13 | ADT<br>F12 | ADT<br>F11 | ADT<br>F10 | ADT<br>F9 | ADT<br>F8 | ADT<br>F7 | ADT<br>F6 | ADT<br>F5 | ADT<br>F4 | ADT<br>F3 | ADT<br>F2 | ADT<br>F1 | ADT<br>F0 |
| Initial value:   | 0          | 0          | 0          | 0          | 0          | 0          | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| R/W: R/(W)*R |            |            |            |            |            |            |           |           |           |           |           |           |           |           |           |           |

Note: \* Only 0 can be written to clear the flag after reading the flag as 1.

|         |           | Initial |        |  |  |  |  |  |  |
|---------|-----------|---------|--------|--|--|--|--|--|--|
| Bit     | Bit Name  | Value   | R/W    | Description  |  |  |  |  |  |
| 15 to 1 | ADTF15 to | All 0   | R/(W)* | Interrupt Conversion End Flag  |  |  |  |  |  |
|         | ADTF1     |         |        |  | ADTFn is a status flag bit which indicates that an interrupt conversion has been completed. 1 must not be written to ADTFn. When ADTFn is set to 1, an ANn interrupt conversion end interrupt (ADIDn) can be generated. See table 19.2 for correspondence between ADTFn and ANn. |  |  |  |  |
|         |           |         |        | 0: Indicates that an interrupt conversion process on ANn is in idle state.     |  |  |  |  |  |
|         |           |         |        |  |  |  |  |  | <ol> <li>Indicates that an interrupt conversion process on<br/>ANn has been completed and the conversion result<br/>has been transferred to ADRn.</li> </ol> |
|         |           |         |        |  | [Clearing condition]   |  |  |  |  |
|         |           |         |        | • 0 is written to this bit after reading 1.                                    |  |  |  |  |  |
|         |           |         |        | [Setting condition]  |  |  |  |  |  |
|         |           |         |        | <ul> <li>An interrupt conversion process on ANn has been completed.</li> </ul> |  |  |  |  |  |

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |
|-----|----------|------------------|--------|---|
| 0   | ADTF0    | 0                | R/(W)* |   |
|     |          |                  |        | ADTF0 is a status flag bit which indicates that an interrupt conversion has been completed. 0 must not be written to ADTF0. When ADTF0 is set to 1, AN0 interrupt conversion end interrupt (ADID0) can be generated. See table 19.2 for correspondence between ADTF0 and AN0. |
|     |          |                  |        | 0: Indicates that an interrupt conversion process on AN0 is in idle state.  |
|     |          |                  |        | 1: Indicates that an interrupt conversion process on AN0 has been completed and the conversion result has been transferred to ADR0.   |
|     |          |                  |        | [Clearing conditions]   |
|     |          |                  |        | • 0 is written to this bit after reading 1.   |
|     |          |                  |        | A-DMAC is started-up by ADID0.  |
|     |          |                  |        | [Setting condition]   |
|     |          |                  |        | <ul> <li>An interrupt conversion process on AN0 has been completed.</li> </ul>  |

Notes: \* Only 0 can be written to clear the flag after reading the flag as 1.

- 1. Even when the ADTFn is not cleared to 0, an interrupt conversion request on ANn can be accepted. Storing timing on an A/D data register n should be provided with care.
- 2. n = 1 to 15

#### ADTRF1

| Bit:      | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|-----------|------------|------------|------------|------------|------------|------------|------------|------------|
|           | ADT<br>F47 | ADT<br>F46 | ADT<br>F45 | ADT<br>F44 | ADT<br>F43 | ADT<br>F42 | ADT<br>F41 | ADT<br>F40 |
| . دماییم، |            | ^          |            |            |            |            |            |            |

R/W: R/(W)\*R/(W)\*R/(W)\*R/(W)\*R/(W)\*R/(W)\*R/(W)\*R/(W)\*

Note: \* Only 0 can be written to clear the flag after reading the flag as 1.

|        |           | Initial |        |   |
|--------|-----------|---------|--------|---|
| Bit    | Bit Name  | Value   | R/W    | Description   |
| 7 to 0 | ADTF47 to | All 0   | R/(W)* | Interrupt Conversion End Flag   |
|        | ADTF40    |         |        | ADTFn is a status flag bit which indicates that an interrupt conversion has been completed. 0 must not be written to ADTFn. When ADTFn is set to 1, an ANn interrupt conversion end interrupt (ADIDn) can be generated. ANn can generate a DMA transfer request to the DMAC. See table 19.2 for correspondence between ADTFn and ANn. |
|        |           |         |        | 0: Indicates that an interrupt conversion process on ANn is in idle state.  |
|        |           |         |        | 1: Indicates that an interrupt conversion process on ANn has been completed and the conversion result has been transferred to ADRn.   |
|        |           |         |        | [Clearing conditions]   |
|        |           |         |        | • 0 is written to this bit after reading 1.   |
|        |           |         |        | A-DMAC is started-up by ADIDn.  |
|        |           |         |        | [Setting condition]   |
|        |           |         |        | An interrupt conversion process on ANn has been completed.  |

Notes: \* Only 0 can be written to clear the flag after reading the flag as 1.

- 1. Even when the ADTFn is not cleared to 0, an interrupt conversion request on ANn can be accepted. Storing timing on an A/D data register n should be provided with care.
- 2. n = 40 to 47

# A/D Interrupt Trigger Conversion Interrupt Enable Registers 0 and 1 (ADTRD0 and ADTRD1)

ADTRD0 and ADIRD1 enable or disable an A/D interrupt conversion end interrupt generation when the ADTF bit in ADTRF is set to 1. ADTRD0 and ADTRD1 are initialized to H'0000 and H'00 respectively by a power-on reset or in hardware standby mode.

#### ADTRD0

| Bit:           | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|----------------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                | ADI  | ADI  | ADI  | ADI  | ADI  | ADI  | ADI | ADI | ADI | ADI | ADI | ADI | ADI | ADI | ADI | ADI |
|                | DE15 | DE14 | DE13 | DE12 | DE11 | DE10 | DE9 | DE8 | DE7 | DE6 | DE5 | DE4 | DE3 | DE2 | DE1 | DE0 |
| Initial value: | 0    | 0    | 0    | 0    | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|                | B/W  | R/W  | B/W  | B/W  | B/W  | B/W  | R/W | B/W | B/M | B/W | B/W | R/W | B/W | B/W | R/W | B/W |

| Bit     | Bit Name   | Initial<br>Value | R/W | Description  |
|---------|------------|------------------|-----|--|
| 15 to 1 | ADIDE15 to | All 0            | R/W | Interrupt Conversion End Interrupt Enable  |
|         | ADIDE1     |                  |     | The ADIDEn bit enables or disables an ANn interrupt conversion end interrupt (ADIDn) to be generated. To prevent incorrect operation, both the ADITACT bit in ADREF0 must be cleared to 0 while the ADIDEn bit value is changed. |
|         |            |                  |     | If the ADIDEn bit is 1 when the ADTFn bit of the interrupt conversion end flag register is set to 1 upon completion of ANn interrupt conversion, the ADIDn signal is generated.  |
|         |            |                  |     | The ADIDn signal can be cleared by clearing ADTFn or ADIDEn to 0.  |
|         |            |                  |     | The correspondence among the ADIDEn bit, ANn, and ADIDn is shown in table 19.2.  |
|         |            |                  |     | <ol> <li>Disables an interrupt request upon completion of<br/>ANn interrupt conversion (ADIDn).</li> </ol>   |
|         |            |                  |     | 1: Enables an interrupt request upon completion of ANn interrupt conversion (ADIDn).   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 0   | ADIDE0   | 0                | R/W | Interrupt Conversion End Interrupt Enable  |
|     |          |                  |     | The ADIDE0 bit enables or disables an AN0 interrupt conversion end interrupt (ADID0) to be generated. To prevent incorrect operation, the ADITACT bit in ADREF0 must be 0 while the ADIDE0 bit is changed. |
|     |          |                  |     | If the ADIDE0 bit is 1 when the ADTF0 bit of the interrupt conversion end flag register is set to 1 upon completion of AN0 interrupt conversion, the ADID0 signal is generated.                            |
|     |          |                  |     | The ADID0 signal can be cleared by clearing ADTF0 or ADIDE0 to 0.  |
|     |          |                  |     | Further, the AN0 can perform a DMA transfer by means of the A-DMAC based on ADID0. See table 19.2 for correspondence between ADIDE0, AN0, and ADID0.   |
|     |          |                  |     | <ol> <li>Disables an interrupt request upon completion of<br/>AN0 interrupt conversion (ADID0) or a DMA<br/>transfer request.</li> </ol>   |
|     |          |                  |     | <ol> <li>Enables an interrupt request upon completion of<br/>AN0 interrupt conversion (ADID0) or a DMA<br/>transfer request.</li> </ol>  |

# ADTRD1

| Bit:           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                | ADI<br>DE47 | ADI<br>DE46 | ADI<br>DE45 | ADI<br>DE44 | ADI<br>DE43 | ADI<br>DE42 | ADI<br>DE41 | ADI<br>DE40 |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| R/W:           | R/W         |

| Bit    | Bit Name   | Initial<br>Value | R/W  | Description   |
|--------|------------|------------------|--|---|
| 7 to 0 | ADIDE47 to | All 0            | R/W  | Interrupt Conversion End Interrupt Enable   |
|        | ADIDE40    |                  | The ADIDEn bit enables or disables an ANn interrupt conversion end interrupt (ADIDn) to be generated. To prevent incorrect operation, the ADITACT bit in ADREF1 must be 0 while the ADIDEn bit is changed. |   |
|        |            |                  |  | If the ADIDEn bit is 1 when the ADTFn bit in the interrupt conversion end flag register is set to 1 upon completion of ANn interrupt conversion, the ADIDn signal is generated. |
|        |            |                  | The ADIDn signal can be cleared by clearing ADTFn or ADIDEn to 0.  |   |
|        |            |                  | Further, the AN40 to AN47 can perform DMA transfers by means of the DMAC based on ADID40 to ADID47. See table 19.2 for correspondence between ADIDEn, ANn, and ADIDn.                                      |   |
|        |            |                  |  | <ol> <li>Disables an interrupt request upon completion of<br/>ANn interrupt conversion (ADIDn) or an DMA<br/>transfer request.</li> </ol>                                       |
|        |            |                  |  | <ol> <li>Enables an interrupt request upon completion of<br/>ANn interrupt conversion (ADIDn) or an DMA<br/>transfer request.</li> </ol>  |

Page 968 of 1824

#### 19.4.13 Interface with CPU

The A/D data register is a 16-bit register. The peripheral bus connected to the CPU is 16 bits. The A/D data register must be read in units of words (16 bits). If the A/D data register is read in byte units by dividing a word into upper and lower bytes and performing read operations twice on it, the A/D converted value read in the first read operation and that read in the second read operation may change. To avoid this error, the A/D data register should not be read in byte units.

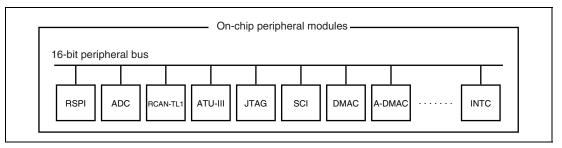


Figure 19.3 Interface between CPU and A/D Converter (ADC)

# 19.5 Operation

#### 19.5.1 Scan Conversion

A scan conversion is performed in two operating modes: the single-cycle scan mode and continuous scan mode. In the single-cycle scan mode, one or more specified channels are scanned once. In the continuous scan mode, one or more specified channels are scanned until the ADST bit is cleared to 0 (changed from 1 to 0) by software.

The single-cycle scan mode is selected by clearing the ADCS bit in ADCSR to 0, while the continuous scan mode is selected by setting the ADCS bit to 1. A scan conversion is performed in ascending order of channel number (ADC A: ANO  $\rightarrow$  AN27, ADC B: AN40  $\rightarrow$  AN48).

In single-cycle scan mode, after scanning all selected channels once, the A/D converter sets the ADF bit in ADREF to 1 and then clears the ADSCACT bit in ADREF to 0 to completes the scan conversion. In continuous scan mode, after scanning all selected channels once, the A/D converter sets the ADF bit to 1 and then continues to scanning. The ADF bit is set to 1 each time scanning on specified channels is completed.

To stop the scanning, write 0 to the ADST bit when it is 1. Writing 0 to the ADST bit when it is 0 does not affect the A/D converter. Similarly, writing 1 to the ADST bit when it is 1 does not affect the A/D converter. Therefore, to stop a scan conversion started by a request other than the ADST bit, first write 1 to the ADST bit and then write 0 to it.

When the ADF bit is set to 1 while the ADIE bit in ADCSR is set to 1, an ADI interrupt request is generated. To clear the ADF bit to 0, write 0 to the ADF bit after reading it as 1. When the DMAC is started by an ADI interrupt, the ADF bit is automatically cleared to 0 and the ADI interrupt is also cleared.

### 19.5.2 Single-Cycle Scan Conversion Mode

The following is an example operation of single-scan conversion where three channels ANO, AN3, and AN9 are selected and an ADI0 interrupt is enabled. The same operations can also apply to ADC B.

- 1. Clear the ADCS bit in A/D control register 0 (ADCSR0) to 0 and set the ADIE bit in ADCSR0 to 1.
- 2. Set bits ANS0, ANS3 and ANS9 in the A/D channel select register 0 (ADANS0).
- 3. Set the ADST bit in A/D control register 0 (ADSCR0) to 1 to start scan conversion. If the ADST bit is already set to 1, write 1 to it after clearing it to 0.
- 4. Starting the scan conversion sets the ADSCACT bit to 1. Then, the A/D conversion on channel AN0 is started. On completion of the A/D conversion, the A/D converted value is transferred to ADR0. After that, channels AN3 and AN9 are scanned in the order in the same way as in AN0.
- 5. When the A/D converted values of all the selected channels (AN0, AN3, and AN9) have been transferred to ADRn, the ADF bit is set to 1. At this time, an ADI0 interrupt is generated since the ADIE bit is set to 1. The ADSCACT bit is cleared to 0 and the scan conversion is completed.
- 6. Next, the ADI0 interrupt handler is started. In the interrupt handler, clear bit ADI0 by writing 0 to the ADF bit after reading it as 1. After that, read the contents of ADR0, ADR3, and ADR9.
- 7. Complete the ADI0 interrupt handler.

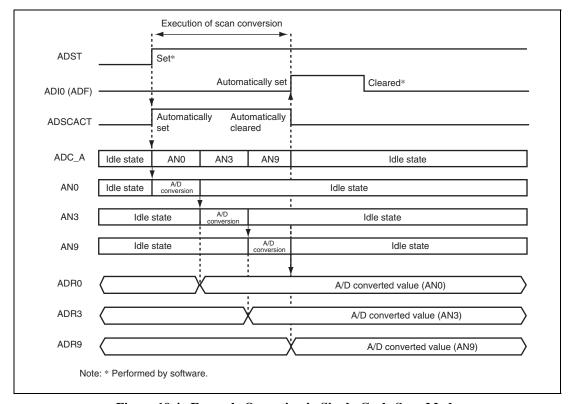


Figure 19.4 Example Operation in Single-Cycle Scan Mode

#### 19.5.3 Continuous Scan Conversion Mode

The following is an example operation of continuous scan conversion where three channels AN0, AN3, and AN9 are selected and an ADI0 interrupt is enabled. The same operations can also apply to ADC B.

- 1 Set the ADCS bit and ADIE bit in A/D control register 0 (ADCSR0) to 1.
- 2. Set bits ANS0, ANS3 and ANS9 bits in A/D channel select register 0 (ADANS0).
- 3. Set the ADST bit in A/D control register 0 (ADSCR0) to 1 to start scan conversion. If the ADST bit is already 1, write 1 to it after clearing it to 0.
- 4. Starting the scan conversion sets the ADSCACT bit to 1. Then, the A/D conversion on channel AN0 is started. On completion of the A/D conversion, the A/D converted value is transferred to ADR0. After that, channels AN3 and AN9 are scanned in the order in the same way as in AN0.
- 5. When the A/D converted values of all the selected channels (AN0, AN3, and AN9) have been transferred to ADRn, the ADF bit is set to 1. At this time, an ADI0 interrupt is generated since the ADIE bit is set to 1. Also, the scan conversion returns to the start.
- 6. The ADI0 interrupt handler is started simultaneously. In the interrupt handler, clear bit ADI0 by writing 0 to the ADF bit after reading it as 1. After that, read the contents of ADR0, ADR3, and ADR9.
- 7. Complete the ADI0 interrupt handler.
- 8. Steps 4 to 7 are repeated as long as the ADST bit is 1. Clearing the ADST bit to 0 clears the ADSCACT bit to 0, and completes the scan conversion. Setting the ADST bit to 1 initiates scan conversion.

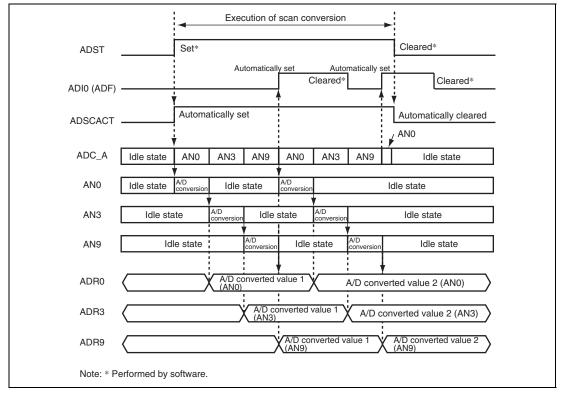


Figure 19.5 Example Operation in Continuous Scan Mode

### 19.5.4 Interrupt Conversion

When an ATU-III timer trigger or software trigger is requested on channels AN0 to AN15 and AN40 to AN47, A/D conversion is performed on the requested channels. Whereas the scan conversion converts all the selected channels for a request, the interrupt conversion converts all the channels selected by a request.

To perform interrupt conversion, set the ADTRGE bit in ADTRE to 1 and select the trigger source by the ADTRS bit in ADTRS. When interrupt conversion is requested by the selected trigger source, A/D conversion is performed on the corresponding AN channel. On completion of the interrupt conversion on the AN channel, the ADTF bit in ADTRF is set to 1. The ADTF bit is set to 1 each time an interrupt conversion is performed on an AN channel. Furthermore, if any interrupt conversion is performed, the ADITACT bit in ADREF is set to 1. When A/D conversion has been completed on all AN channels to which interrupt conversion is requested, the ADITACT bit is cleared to 0.

When interrupt conversion requests conflict, A/D conversion is performed according to the priority. ADC\_A is prioritized as AN0 > AN1 > ... AN14 > AN15, that is, the lower channel number corresponds to the higher priority. ADC\_B is prioritized as AN40 > AN44 > AN41 > AN45 > AN45 > AN46 > AN43 > AN47, thus channel AN40 given the highest priority, and AN47 given the lowest priority. Note that this priority in the ADC\_B channels is not in ascending order of channel numbers. When interrupt conversion is requested on other channels (ANj and ANk) during the interrupt conversion on channel ANi, the A/D conversion is not interrupted during the conversion regardless of the priority. In this case, on completion of the A/D conversion on channel ANi, A/D conversion is performed according to the priority on remaining channels (in this case, ANj and ANk) in which interrupt conversion requests are pending. Therefore, the priority on interrupt conversion determines which channel is to be converted for the next operation. When a single trigger source generates interrupt conversion requests on two channels or multiple trigger sources simultaneously generate interrupt conversion requests, A/D conversion is performed according to this priority.

When interrupt conversion is requested during scan conversion, the scan conversion on channel ANi is suspended, and A/D conversion on the other channel (ANj) in which the interrupt conversion was requested is performed. On completion of the interrupt conversion on channel ANj, the scan conversion is resumed from the interrupted channel (ANi). This scheme ensures that the length of time required from the initiation of an interrupt conversion request to the completion of it is always constant. This makes it possible, for example, to perform A/D conversion in pinpoint accuracy by synchronizing them with the operation of A/D conversion sources that are external to the LSI.

When the ADTF bit is set to 1 while the ADIDE bit in the ADTRD register is set to 1, an ADID interrupt is requested. To clear the ADTF bit to 0, write 0 to the ADTF bit after reading it as 1. If the DMAC or A-DMAC is started by an ADID interrupt, note that the ADTF bit is automatically cleared to 0 and the ADID interrupt is also cleared. The DMA transfer of the DMAC is supported on channels AN40 (ADID40) to AN47 (ADID47); and the DMA transfer of the A-DMAC is supported on channel AN0 (ADID0).

### 19.5.5 Example Operation of Interrupt Conversion

The following is an example operation of interrupt conversion where timer G0 is selected as a trigger source for channel AN40 and timer G2 is selected as a trigger source for channels AN42 and AN46.

- 1. Set bits ADTRGE40, ADTRGE42, and ADTRGE46 in A/D interrupt trigger enable register 1 (ADTRE1) to 1.
- 2. Clear bits ADTRS40, ADTRS42, and ADTRS46 in A/D interrupt trigger source select register 1 (ADTRS1) to 0.
- 3. Subsequently, interrupt conversion requests are generated by timers G0 and G2 at intervals specified by the ATU-III registers. For details on the ATU-III registers, see section 13, Advanced Timer Unit III (ATU-III).
- 4. When interrupt conversion is requested by timer G0, the ADITACT bit is set to 1 and interrupt conversion on channel AN40 is performed. On completion of A/D conversion on AN40, the A/D converted value of AN40 is transferred to ADR40 and bit ADTF40 in ADTRF1 is set to 1. The ADITACT bit is cleared to 0 and the interrupt conversion is completed. Furthermore, if bit ADIDE40 is 1, an ADID40 interrupt is requested to the CPU.
- 5. When interrupt conversion is requested by timer G2, the ADITACT bit is set to 1, and interrupt conversion on channels AN42 and AN46 is performed. Then A/D conversion on channel AN42 is performed. On completion of the conversion, the A/D converted value of AN42 is transferred to ADR42, and bits ADTF42 in ADTRF1 is set to 1. An A/D conversion on channel AN46 is then performed. On completion of the conversion, the A/D converted value of AN46 is transferred to ADR46, and bit ADTF46 in ADTRF1 is set to 1. The ADITACT bit is cleared to 0 and the interrupt conversion is completed. Further, if bits ADIDE42 and ADIDE46 in ADTRD1 are set to1 when either bit ADTF42 or ADTF46 is set to 1, the A/D converter requests an ADID42 or an ADID46 interrupt to the CPU.
- 6. Subsequently, steps 4 to 5 are repeated. The following is an example operation when requests by timers G0 and G2 conflict.

#### (1) Example Operation 1

When a timer G0 interrupt conversion request is input during the A/D conversion on channel AN42 due to a timer G2 interrupt conversion request, the request is processed as follows.

The timer G0 interrupt source is retained in the A/D converter until the conversion on channel AN42 is completed. On completion of the A/D conversion on channel AN42, A/D conversion on channels AN40 and AN46 is performed in the order according to the priority.

#### (2) Example Operation 2

When interrupt conversion requests by timers G0 and G2 are input simultaneously, the requests are processed as follows.

The timer G0 and timer G2 interrupt sources are retained in the A/D converter. A/D conversion on channels AN40, AN42, and AN46 in the order according to the priority.

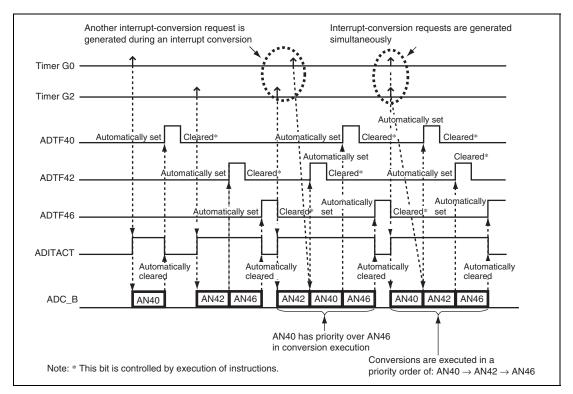


Figure 19.6 Example Operation of Interrupt Conversion

#### 19.5.6 Interrupt Conversion during Scan Conversion

The following is an example operation where single scan conversion on three channels ANO, AN3, and AN9 is started by a scan conversion request from timer G4 and then an interrupt conversion on channel AN6 is started by a interrupt conversion request from timer D03A.

- 1. Clear the ADCS and EXTRG bits in A/D control register 0 (ADCSR0) to 0, and set the TRGE bit in ADCSR0 to 1.
- 2. Set bits ANS0, ANS3 and ANS9 bits in A/D channel select register 0 (ADANS0) to 1.
- 3. Set bit ADTRGE6 in A/D interrupt trigger enable register 0 (ADTRE0) to 1.
- 4. Clear bit ADTRS6 in A/D interrupt trigger source select register (ADTRS0) to 0.
- 5. Subsequently, an scan conversion request is generated by timer G4 and an interrupt conversion request is generated by timer D03A at intervals specified by the ATU-III registers. For details on the ATU-III registers, see section 13, Advanced Timer Unit III (ATU-III).
- 6. When scan conversion is requested by timer G4, the ADSCACT bit is set to 1. Then, A/D conversion on channels AN0, AN3, and AN9 is performed in the order. On completion of the conversion, the ADF bit is set to 1 and the ADSCACT bit is cleared to 0, indicating that the scan conversion is completed.
- 7. When interrupt conversion is requested by timer D03A, the ADITACT bit is set to 1 and interrupt conversion on channel AN6 is performed. On completion of the A/D conversion on channel AN6, bit ADTF6 in ADTRF0 is set to 1 and the ADITACT bit is cleared to 0, indicating that the interrupt conversion is completed.
- 8. Subsequently, steps 6 to 7 are repeated. The following is an example operation where a scan conversion and an interrupt conversion conflict.

### (1) Example Operation

When a timer D03A interrupt conversion request is input during the A/D conversion on channel AN3 in the scan conversion due to a timer G4 scan conversion request, the request is processed as follows.

The timer D03A interrupt source is retained in the A/D converter, and the scan conversion on channel AN3 is suspended. The priority is applied to channels AN3 and AN9 on which scan conversion is pending, and is applied to AN6 which is the current request. In this case, the A/D conversion on channels AN6, AN3, and AN9 in the order.

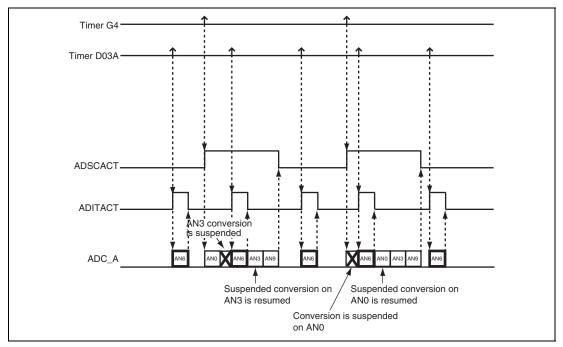


Figure 19.7 Operation Example of Interrupt Conversion during Scan Conversion

### 19.5.7 Analog Input Sampling and Scan Conversion Time

The A/D converter includes sample and hold circuit. When start-of-scan-conversion delay time ( $t_p$ ) have passed after the ADST bit in ADCSR is set to 1, the A/D converter samples the analog input, and then begins the conversion process.

Figure 19.8 shows a timing chart for a scan conversion on one channel in single-cycle scan mode. Scan conversion time  $(t_{SCAN})$  includes start-of-scan-conversion delay time  $(t_{p})$ , analog input sampling time  $(t_{SPL})$ , A/D conversion processing time  $(t_{CONV})$  and end-of-scan-conversion delay time  $(t_{ED})$ . The scan conversion time is shown in table 19.5.

The scan conversion time  $(t_{SCAN})$  in single-cycle scan mode for which the number of selected channels is n can be determined according to the following equation:

$$t_{SCAN} = t_D + \{(t_{SPI} + t_{CONV}) \times n\} + t_{FD}$$

The scan conversion time for the first cycle in continuous scan mode is  $t_{\text{\tiny SCAN}}$  for single-cycle scan minus  $t_{\text{\tiny FD}}$ .

The scan conversion time for the second and subsequent cycles in continuous scan mode is a fixed time, which is equal to  $\{(t_{SPI} + t_{CONV}) \times n\}$ .

Table 19.5 Scan Conversion Time

|                                     |                   | Based on Pφ         |                         |              |
|-------------------------------------|-------------------|---------------------|-------------------------|--------------|
| Item                                | Symbol            | Low Speed (CKS = 0) | High Speed<br>(CKS = 1) | <br>Unit     |
| Start-of-scan-conversion delay time | t <sub>D</sub>    | 7                   | 5                       | State        |
| Analog input sampling time          | t <sub>SPL</sub>  | 20                  | 10                      |              |
| A/D conversion processing time      | t <sub>conv</sub> | 30                  | 15                      | _            |
| End-of-scan-conversion delay time   | t <sub>ED</sub>   | 4                   | 2                       | _            |
| Scan conversion time                | t <sub>scan</sub> | 61                  | 32                      | <del>_</del> |

Note: CKS cannot be set to 1 when two-time multiplication peripheral clock is set.

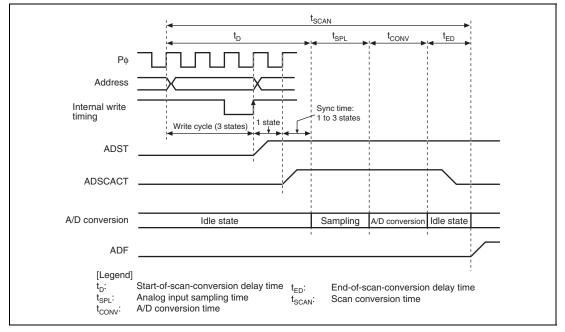


Figure 19.8 Timing Diagram for Scan Conversion (Single Channel, Single Cycle)

### 19.5.8 Starting Scan Conversion with External Trigger

The A/D converter can be activated by the input of an external trigger. To start up the A/D converter by an external trigger, the pin function should be set up using the pin function controller (PFC). After applying a high-level signal to the ADTRG pin, both the TRGE and EXTRG bits in the A/D control register (ADCSR) should be set to 1. If a low-level signal is then input to the ADTRG pin, the A/D converter detects a pulse fall edge and sets the ADSCSCT bit to 1.

Figure 19.9 shows an external of trigger input timing. If the low-speed is selected (CKS = 0), the timing at which the ADSCACT bit is set to 1 is 4 states after the falling edge of the ADTRG pin is sampled; while if the high-speed is selected (CKS = 1), it is 3 states after the fall edge of the ADTRG pin is sampled.

The timing at which a scan conversion is started after the ADSCACT bit is set to 1 is the same as the case where the ADST bit is set to 1 from 0 by software. For details on pin function setting, see section 22, Pin Function Controller (PFC).

To stop the scan conversion process while it is in progress, write 1 to the ADST bit and then write 0 to it.

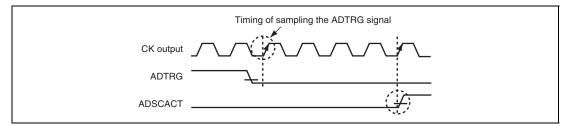


Figure 19.9 External Trigger Input Timing

### 19.5.9 Starting Scan Conversion with ATU-III Timer Trigger

A scan conversion can be activated by an ATU-III timer trigger. To start up a scan conversion by an ATU-III timer trigger, both the TRGE and EXTRG bits in the A/D control register (ADCSR) should be cleared to 0. If a timer trigger (timerG4 or timer G5) is entered in this situation, the ADSDACT bit is set to 1. The timing at which a scan conversion is started after the ADSCACT bit is set to 1 is the same as the case where the ADST bit is set to 1 from 0 by software.

To stop the scan conversion process while it is in progress, write 1 to the ADST bit and then write 0 to it.

### 19.5.10 Monitoring via ADEND\_A and ADEND\_B Output Pins

The timing at which AN0 and AN40 are scan-converted can be monitored via the ADEND\_A and ADEND\_B output pins, respectively. For details on pin function setting, see section 22, Pin Function Controller (PFC).

Figure 19.10 shows ADEND\_A and ADEND\_B output examples. If ADEND\_A and ADEND\_B outputs are selected by the PFC, monitor signals are output, respectively, from the ADEND\_A and ADEND\_B output pins during the conversion processing of channels AN0 and AN40. Upon completion of the AN0 and AN40 samplings, outputs are produced from the ADEND\_A and ADEND\_B pins, respectively.

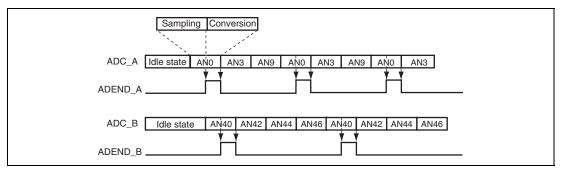


Figure 19.10 Example of ADEND A and ADEND B Outputs

Note: If an interrupt conversion is performed while high-level signals are output from the ADEND\_A and ADEND\_B pins, low-level signals are output once from these pins. After that, because channels AN0 and AN40 are converted again in a scan conversion process, high-level signals are output from the ADEND\_A and ADEND\_B pins again. In addition, if channels AN0 and AN40 are converted in an interrupt conversion process, high-level signals are also output from the ADEND\_A and ADEND\_B pins. Further, if channels AN0 and AN40 are set to A/D-converted value addition mode, high-level signals are output from the ADEND\_A and ADEND\_B pins only during the final A/D conversion (in the 4th conversion if four-addition conversion is performed, for example).

### 19.6 Interrupt Sources and DMA Transfer Request

#### 19.6.1 Interrupt Requests on Completion of Scan Conversion

The A/D converter can generate a scan conversion end interrupt request (ADI) to the CPU. By setting the ADIE bit in the A/D control register (ADCSR) to 1, an ADI interrupt is enabled; by clearing the bit to 0, an ADI interrupt is disabled. In addition, the DMAC can be started up when an ADI interrupt is generated. In this case, interrupts are not generated to the CPU. If the DMAC is started upon an ADI interrupt, the ADF bit in the A/D conversion status register (ADREF) is automatically cleared to 0 when data transfer is performed by the DMAC.

For details on DMAC settings, see section 11, Direct Memory Access Controller (DMAC).

Note: The ADF bit is not cleared by an interrupt request to the CPU.

#### 19.6.2 Interrupt Requests on Completion of Interrupt Conversion

The A/D converter can generate interrupt conversion end interrupt requests (ADID0 to ADID15, ADID40 to ADID47) to the CPU upon completion of an interrupt conversion. By setting the ADIDE0 to ADIDE15 and ADIDE40 to ADIDE47 bits in the A/D interrupt trigger processing end interrupt enable register (ADTRD) to 1, the ADID0 to ADID15 and ADID40 to ADID47 interrupts are enabled, respectively; by clearing the bits to 0, the ADID0 to ADID15 and ADID40 to ADID47 interrupts are disabled, respectively. If the DMAC is activated by an ADID40 to ADID47 interrupt, a corresponding bit in the ADTF40 to ADTF47 bits in the A/D interrupt trigger processing end flag register 1 (ADTRF1) is automatically cleared to 0 when data transfer is performed by the DMAC. If the A-DMAC is activated by an ADID0 interrupt, the ADTF0 bit in the A/D interrupt trigger processing end flag register 0 (ADTRF0) is automatically cleared to 0 when data transfer is performed by the ADMAC.

For details on DMAC settings, see section 11, Direct Memory Access Controller (DMAC) and section 12, Automotive Direct Memory Access Controller (A-DMAC).

Note: The ADTF bit is not cleared by an interrupt request to the CPU.

# 19.7 Definition of A/D Conversion Accuracy

The definition of A/D conversion accuracy is described below.

Resolution

This indicates the number of digital output codes in the A/D converter

• Quantization error

This error, which is inherent to the A/D converter, is given as 1/2LSB (figure 19.11).

Offset error

This error, which is exclusive of quantization error, is a deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from a minimum voltage value B'0000000000000 to B'000000000001 (figure 19.11).

Full scale error

This error, which is exclusive of quantization error, is a deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from B'111111111110 to B'1111111111111 (figure 19.11).

• Nonlinearity error

This error, which is exclusive of offset error, full scale error and quantization error, is a deviation from the ideal A/D conversion characteristics through the zero-scale and full-scale transitions (figure 19.11).

Absolute accuracy

This is a deviation of the digital value from the analog input value. This includes offset error, full scale error, quantization error, and nonlinearity error.

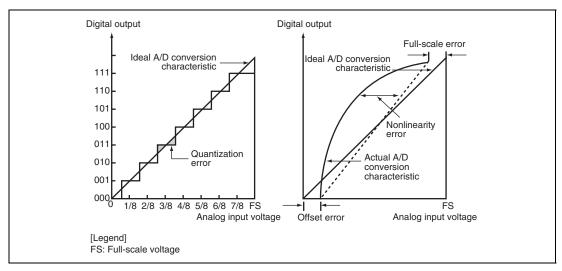


Figure 19.11 Definition of A/D Conversion Accuracy

# 19.8 Usage Notes

### 19.8.1 Analog Input Voltage Range

The voltage applied to an analog input pin (ANn) during A/D conversion should be within the following range:

$$AVrefl_A \le ANn (n = 0 \text{ to } 27, 40 \text{ to } 48) \le AVrefl_A$$

$$AVrefl_B \le ANn (n = 0 \text{ to } 27, 40 \text{ to } 48) \le AVrefl_B$$

### 19.8.2 Relationship among $AV_{CC}$ , $AV_{SS}$ , $V_{CC}$ , and $V_{SS}$

When using the A/D converter, make sure that the following relationships are held among  $AV_{cc}$ ,  $AV_{ss}$ ,  $V_{cc}$  and  $V_{ss}$ :

$$AV_{cc} = 5.0 V \pm 0.5 V$$
,  $AV_{ss} = V_{ss}$ 

When the A/D converter is not used,  $AV_{cc}$  pin must not be open. In this case, the following relationship should be held between  $AV_{ss}$  and  $V_{ss}$ :

$$AV_{SS} = V_{SS}$$

# 19.8.3 Allowable Settings for Pins AVrefh\_A/AVrefh\_B and AVrefl\_A/AVrefl\_B

The allowable settings for the AVrefh\_A/AVrefh\_B pins are as follows:

When the A/D converter is used:  $AVrefh_A = 4.5V$  to  $AV_{cc}$ ,  $AVrefh_B = 4.5V$  to  $AV_{cc}$ 

When the A/D converter is not used: AVrefh\_A  $\leq$  AV<sub>cc</sub>, AVrefh\_B  $\leq$  AV<sub>cc</sub>

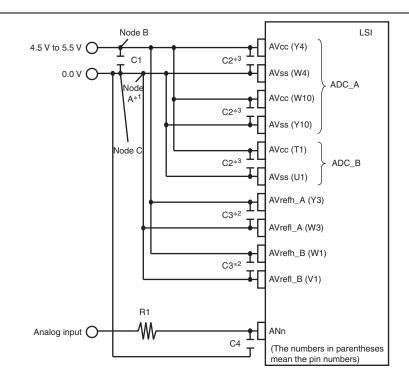
If any value outside the above range is set, it can adversely affect the reliability of the LSI. For the  $AVrefl_A/AVrefl_B$  pin, set  $AVrefl_A/AVrefl_B = AV_{ss} = V_{ss}$ .

### 19.8.4 Precautions on Board Design

For designing a board, to the maximum extent possible the digital circuits should be laid out separately from the analog circuits. Layouts involving the crossing of signal lines for digital circuits and signal lines for analog circuits, or placing them in proximity to each other, should be avoided. If the dissimilar signal lines are placed in close proximity to each other, the resulting induction can lead to a malfunction of the analog circuits or produce an adverse impact on A/D conversion values.

It should be noted that the analog input pins (AN0 to AN27, AN40 to AN48), the analog reference voltages (AVrefh\_A/AVrefh\_B, AVrefl\_A/AVrefl\_B), and the analog power supply (AV $_{cc}$ ) should be isolated from the digital circuits by means of analog grounding (AV $_{ss}$ ). In addition, the analog ground (AV $_{ss}$ ) should be connected in one point to a stable digital ground (V $_{ss}$ ) on the board.

As the protection circuit to prevent the analog input pins (AN0 to AN27, AN40 to AN48) from damages, by such abnormal voltages as excessive surges, bypass capacitors should be connected between  $AV_{cc}$  and  $AV_{ss}$ , and also between  $AVrefh\_A/B$  and  $AVrefl\_A/B$ , as illustrated in figures 19.12 (1) to 19.12 (3). Also a filter capacitor connected to an analog input pin (ANn) should be connected to the  $AV_{ss}$ . The capacitance values of the bypass capacitors connected between  $AV_{cc}$  and  $AV_{ss}$  or between  $AVrefh\_A/B$  and  $AVrefl\_A/B$ , as illustrated in figures 19.12 (1) to 19.12 (3), are reference values. Therefore, for designing a board, care must be taken to choose appropriate capacitance values. In addition, connecting a filter capacitor, as illustrated in figures 19.12 (1) to 19.12 (3), can cause an error by averaging the input currents to analog input pins (ANn). Therefore, care must be taken to choose appropriate circuit constants. Figure 19.12 (1) shows an example of connecting power supples in a basic configuration. Figure 19.12 (2) shows an example of connecting power supples in a configuration that is less affected by the board. Figure 19.12 (3) shows an example of connecting power supples in a configuration in which the voltages of  $AV_{cc}$  and AVrefh can differ from each other.



Reference values: C1 : 10.0  $\mu F$  to 100.0  $\mu F$  (47  $\mu F^{*5}$ )

C2, C3 : 0.1  $\mu$ F to 1.0  $\mu$ F (0.1  $\mu$ F\*5)

 $\begin{array}{cc} \text{C4} & : 0.1 \ \mu\text{F} \\ \text{R1} & : 3 \ k\Omega \end{array}$ 

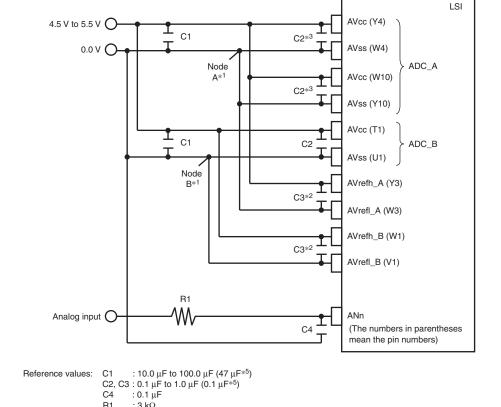
Notes: 1. For designing a board, to reduce the potential difference between AVss and AVrefl, that may cause an offset error, the all wiring impedances between node A and AVss (W4), between node A and AVss (Y10), and between node A and AVrefl\_A, must be the same value that is as low as possible. Similarly, the both wiring impedances between node A and AVss (U1) and between node A and AVrefl\_B must be the same value that is as low as possible.

If an error is getting worse toward the full-scale voltage, the insufficient stability of the voltage of AVrefh\_A/B may cause the problem.

To improve the stability of the voltage, either reduce the wiring impedance of AVrefh\_A/B, or increase the capacitance value of C3.

- 3. It is recommended that the C2 capacitors are connected between the respective pairs of AVcc and AVss. However, when the characteristics can be satisfied by the board design, with concerning the condition described in note 1, one C2 capacitor can be connected in common.
- 4. If an error is getting worse by the simultaneous operation of ADC\_A and ADC\_B, the mutual interference between ADC\_A and ADC\_B must be reduced. Thus, to reduce it, set the connecting node of AVcc (Y4, W10), AVcc (T1), AVrefh\_A and AVrefh\_B, closer to node B. Similarly, set the connecting node of AVss (W4, Y10), AVss (U1), AVrefl\_A and AVrefl\_B, closer to node C. In addition, increasing the capacitance value of C1 is also effective.
- The capacitance values have been evaluated on a Renesas evaluation board. The most appropriate capacitance values should be evaluated on the actual board of the user.

Figure 19.12 (1) Example of Connecting Analog Power Supples and Analog Input Pins (Basic Configuration)

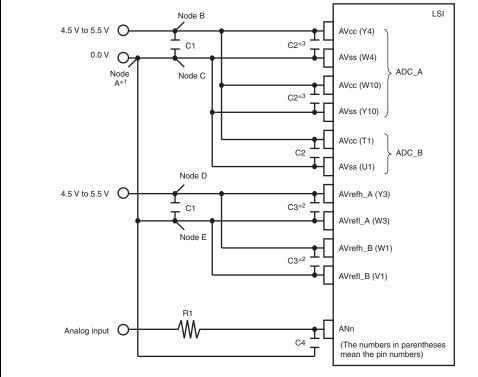


R1 : 3 kΩ

Notes: 1. For designing a board, to reduce the potential difference between AVss and AVrefl, that may cause an offset error, the all wiring impedances between node A and AVss (W4), between node A and AVss (Y10), and between node A and AVrefl\_A, must be the same value that is as low as possible. Similarly, the both wiring impedances between node B and AVss (U1) and between node B and AVrefl B must be the same value that is as low as possible.

- 2. If an error is getting worse toward the full-scale voltage, the insufficient stability of the voltage of AVrefh\_A/B may cause the problem.
  - To improve the stability of the voltage, either reduce the wiring impedance of AVrefh\_A/B, or increase the capacitance value of C3.
- 3. It is recommended that the C2 capacitors for ADC\_A are connected between the respective pairs of AVcc and AVss. However, when the characteristics can be satisfied by the board design, with concerning the condition described in note 1, one C2 capacitor can be connected in common.
- 4. If an error is getting worse by the simultaneous operation of ADC\_A and ADC\_B, increase the capacitance value of C1 to reduce the mutual interference between ADC\_A and ADC\_B.
- 5. The capacitance values have been evaluated on a Renesas evaluation board. The most appropriate capacitance values should be evaluated on the actual board of the user.

Figure 19.12 (2) Example of Connecting Analog Power Supples and Analog Input Pins (Less Affected by Board)



Reference values: C1 : 10.0  $\mu$ F to 100.0  $\mu$ F (47  $\mu$ F\*6)

C2, C3 : 0.1 μF to 1.0 μF (0.1 μF\*6)

 $\begin{array}{ccc} \text{C4} & : 0.1 \ \mu\text{F} \\ \text{R1} & : 3 \ k\Omega \end{array}$ 

Notes: 1. For designing a board, to reduce the potential difference between AVss and AVrefl, that may cause an offset error, the all wiring impedances between node A and AVss (W4), between node A and AVss (Y10), and between node A and AVrefl\_A, must be the same value that is as low as possible. Similarly, the both wiring impedances between node A and AVss (U1) and between node A and AVrefl\_B must be the same value that is as low as possible.

If an error is getting worse toward the full-scale voltage, the insufficient stability of the voltage of AVrefh\_A/B may cause the problem.

To improve the stability of the voltage, either reduce the wiring impedance of AVrefh\_A/B, or increase the capacitance value of C3.

- 3. It is recommended that the C2 capacitors for ADC\_A are connected between the respective pairs of AVcs and AVss. However, when the characteristics can be satisfied by the board design, with concerning the condition described in note 1, one C2 capacitor can be connected in common.
- 4. If an error is getting worse by the simultaneous operation of ADC\_A and ADC\_B, the mutual interference between ADC\_A and ADC\_B must be reduced. Thus, to reduce it, set the connecting node of AVcc (Y4, W10) and AVcc (T1) closer to node B. Similarly, set the connecting node of AVss (W4, Y10) and AVss (U1) closer to node C, the node of AVreff\_A and AVreff\_B to node D, the node of AVreff\_A and AVreff\_B to node E, respectively. In addition, increasing the capacitance value of C1 is also effective.
- 5. By separating the power supplies to AVcc and AVrefh from each other, the voltages of AVcc and AVrefh\_A/B can differ from each other.
  - However, for designing a board, AVrefh\_A/B = 4.5V to AVcc must be satisfied and the ground must be in common, with concerning the condition described in note 1.
- The capacitance values have been evaluated on a Renesas evaluation board. The most appropriate capacitance values should be evaluated on the actual board of the user.

Figure 19.12 (3) Example of Connecting Analog Power Supples and Analog Input Pins (AVcc and AVrefh A/B Can Differ from Each Other)

SH72546R Section 20 JTAG Interface

# Section 20 JTAG Interface

This LSI includes the JTAG interface, providing the boundary scan function conforming to the IEEE standard 1149.1.

### 20.1 Features

- Five test signals: TCK, TDI, TDO, TMS, and TRST
- TAP controller
- Four registers:

Instruction register (SDIR), ID register (SDID), bypass register (SDBPR), and boundary scan register (SDBSR)

Six commands conforming to the IEEE standard 1419.1:
 BYPASS, EXTEST, SAMPLE/PRELOAD, CLAMP, HIGHZ, and IDCODE

Figure 20.1 shows a block diagram of the JTAG interface.

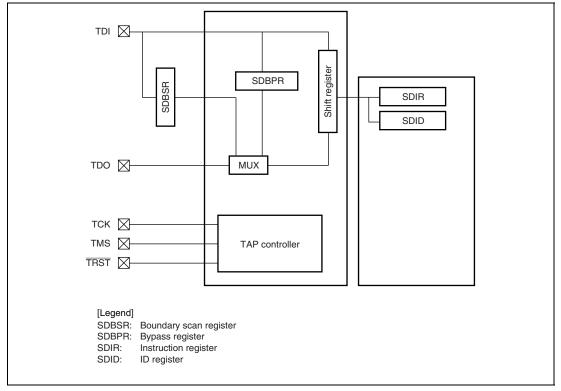


Figure 20.1 Block Diagram of JTAG Interface

# 20.2 Input/Output Pins

Table 20.1 shows the pin configuration of the JTAG interface.

**Table 20.1 Pin Configuration** 

| Pin Name | Description   |
|----------|---|
| TCK      | Serial data input/output clock pin  |
|          | Data is supplied via the data input pin (TDI) and output from the data output pin (TDO) synchronously with the serial data input/output clock signal.   |
| TMS      | Mode select input pin   |
|          | The state of the TAP controller is determined by changing the level of this signal synchronously with the TCK signal. For the protocol, see figure 20.2.  |
| TRST     | Reset input pin   |
|          | The input signal to this pin is accepted asynchronously with the TCK signal; a low-level input signal resets the JTAG interface. This pin must be held low for a specified time after turning on the power whether the JTAG interface is used or not. |
| TDI      | Serial data input pin   |
|          | Data is sent to the JTAG interface by changing the level of this pin synchronously with the TCK signal.   |
| TDO      | Serial data output pin  |
|          | Data is read out from the JTAG interface by reading the level of this pin synchronously with the TCK signal.  |

### 20.3 Register Descriptions

The JTAG interface has the following registers. No register can be accessed by the CPU.

**Table 20.2 Register Configuration** 

| Register Name          | Abbreviation | Initial Value* <sup>1</sup>                  | Size |
|------------------------|--------------|--|------|
| Instruction register   | SDIR         | H'4  | 4    |
| ID register            | SDID         | R5F72546R: H'08085447                        | 32   |
|                        |              | R5E72556R/R5E72546R/R5E72531R:<br>H'0805F447 |      |
| Bypass register        | SDBPR        | Undefined* <sup>2</sup>                      | 1    |
| Boundary scan register | SDBSR        | Undefined                                    | _    |

Notes: \*1 The registers are initialized while the TAP controller is in the Test-Logic-Reset state.

Commands and data can be input to SDIR and SDDR via the serial data input pin (TDI) using serial transfers. Data can be output from SDIR and SDDR via the serial data output pin (TDO). SDBPR, which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS, CLAMP, and HIGHZ modes. SDBSR, which is a 457-bit boundary scan register, is connected between the TDI and TDO pins in SAMPLE/PRELOAD and EXTEST modes. SDID, which is a 32-bit register, can output the fixed code via the TDO pin in IDCODE mode.

Table 20.3 shows the possible serial transfers for the JTAG interface registers.

Table 20.3 Serial Transfers for JTAG Interface Registers

| Register Abbreviation | Serial Input  | Serial Output  |
|-----------------------|---------------|----------------|
| SDIR                  | Available     | Not available* |
| SDBPR                 | Available     | Available      |
| SDBSR                 | Available     | Available      |
| SDID                  | Not available | Available      |

Notes: \* A fixed value (B'1101) is read.

<sup>\*2</sup> Cleared in the Capture-DR state of the BYPASS instruction.

SH72546R Section 20 JTAG Interface

#### 20.3.1 Instruction Register (SDIR)

SDIR is a 4-bit register that holds one of the boundary scan commands. SDIR is initialized when the TRST pin is asserted or the TAP controller is in the Test-Logic-Reset state. If any reserved command is set in SDIR, the BYPASS instruction will be executed.

**Table 20.4 Boundary Scan Commands** 

|          | Co            | mmand Code | s | Description                 |
|----------|---------------|------------|---|-----------------------------|
| 0        | 0             | 0          | 0 | JTAG EXTEST                 |
| 0        | 0             | 0          | 1 | JTAG SAMPLE/PRELOAD         |
| 0        | 1             | 1          | 0 | JTAG CLAMP                  |
| 0        | 1             | 1          | 1 | JTAG HIGHZ                  |
| 0        | 1             | 0          | 0 | JTAG IDCODE (initial value) |
| 1        | 1             | 1          | 1 | JTAG BYPASS                 |
| Other th | nan the above |            |   | Reserved                    |

### 20.3.2 ID Register (SDID)

SDID is a 32-bit register that holds the LSI's ID code. SDID can be read out via the JTAG interface pins when the IDCODE command is set and cannot be written.

The read value is H'08085447 (R5F72546R) or H'0805F447 (R5E72556R, R5E72546R, and R5E72531R); however, the upper four bits may be changed as the LSI version is revised.

### 20.3.3 Bypass Register (SDBPR)

SDBPR is a 1-bit register and is connected between the TDI and TDO pins when SDIR is set to BYPASS mode. The value in SDBPR is not initialized by a power-on reset or  $\overline{TRST}$  pin assertion, but the first value read by the BYPASS instruction is always 0 because this register is initialized in the Capture-DR state of this instruction.

### 20.3.4 Boundary Scan Register (SDBSR)

SDBSR is a shift register arranged on a pad to control the external input and output pins. SDBSR is connected between the TDI and TDO pins when SDIR is set to SAMPLE/PRELOAD or EXTEST mode. The value in SDBSR is not initialized even by a power-on reset or TRST pin assertion and stays undefined.

Table 20.5 shows the relationship between the pins of this LSI and the SDBSR bits.

Table 20.5 Relationship between Pins and SDBSR Bits

| Pin No. | Pin Name                | Input/Output   | Bit No. |
|---------|-------------------------|----------------|---------|
|         | From TDI                |                |         |
| C15     | NMI                     | Input          | 456     |
| B16     | MD_CLK1                 | Input          | 455     |
| D14     | MD_CLK0                 | Input          | 454     |
| B15     | MD_CLKP                 | Input          | 453     |
| C13     | MD3                     | Input          | 452     |
| A11     | FWE                     | Input          | 451     |
| B11     | WDTOVF                  | Output enabled | 450     |
|         |                         | Output         | 449     |
| C11     | PD13/TCLKB/TIJ1         | Output enabled | 448     |
|         |                         | Output         | 447     |
|         |                         | Input          | 446     |
| D10     | PD12/TCLKA/TIOC41/TIJ0  | Output enabled | 445     |
|         |                         | Output         | 444     |
|         |                         | Input          | 443     |
| A9      | PD11/TIOC23/TIF2B/TOE51 | Output enabled | 442     |
|         |                         | Output         | 441     |
|         |                         | Input          | 440     |
| C10     | PD10/TIOC22/TIF1B/TOE50 | Output enabled | 439     |
|         |                         | Output         | 438     |
|         |                         | Input          | 437     |

| Pin No. | Pin Name                | Input/Output   | Bit No. |
|---------|-------------------------|----------------|---------|
| B9      | PD9/TIOC21/TIF0B/TOE43  | Output enabled | 436     |
|         |                         | Output         | 435     |
|         |                         | Input          | 434     |
| A8      | PD8/TIOC20/TIOC33/TOE53 | Output enabled | 433     |
|         |                         | Output         | 432     |
|         |                         | Input          | 431     |
| A7      | PD7/TIOC13/TOE42        | Output enabled | 430     |
|         |                         | Output 429     |         |
|         |                         | Input          | 428     |
| A6      | PD6/TIOC12/TOE41        | Output enabled | 427     |
|         |                         | Output         | 426     |
|         |                         | Input          | 425     |
| B8      | PD5/TIOC11/TOE23/TOE40  | Output enabled | 424     |
|         |                         | Output         | 423     |
|         |                         | Input          | 422     |
| C9      | PD4/TIOC10/TIOC32/TOE52 | Output enabled | 421     |
|         |                         | Output 420     | 420     |
|         |                         | Input          | 419     |
| A5      | PD3/TIOC03/TOE22/TOE53  | Output enabled | 418     |
|         |                         | Output         | 417     |
|         |                         | Input          | 416     |
| A4      | PD2/TIOC02/TOE21/TOE52  | Output enabled | 415     |
|         |                         | Output         | 414     |
|         |                         | Input          | 413     |
| B7      | PD1/TIOC01/TOE20        | Output enabled | 412     |
|         |                         | Output         | 411     |
|         |                         | Input          | 410     |
| B6      | PD0/TIOC00/TIOC31       | Output enabled | 409     |
|         |                         | Output         | 408     |
|         |                         | Input          | 407     |

| Pin No. | Pin Name        | Input/Output   | Bit No. |
|---------|-----------------|----------------|---------|
| D8      | PE13/TOE13      | Output enabled | 406     |
|         |                 | Output         | 405     |
|         |                 | Input          | 404     |
| C8      | PE12/TOE12      | Output enabled | 403     |
|         |                 | Output         | 402     |
|         |                 | Input          | 401     |
| А3      | PE11/TOE11      | Output enabled | 400     |
|         |                 | Output         | 399     |
|         |                 | Input          | 398     |
| A2      | PE10/TOE10      | Output enabled | 397     |
|         |                 | Output         | 396     |
|         |                 | Input          | 395     |
| C7      | PE9/TOE03       | Output enabled | 394     |
|         |                 | Output         | 393     |
|         |                 | Input          | 392     |
| B5      | PE8/TOE02       | Output enabled | 391     |
|         |                 | Output         | 390     |
|         |                 | Input          | 389     |
| D6      | PE7/TOE01/CRx_B | Output enabled | 388     |
|         |                 | Output         | 387     |
|         |                 | Input          | 386     |
| B4      | PE6/TOE00/CTx_B | Output enabled | 385     |
|         |                 | Output         | 384     |
|         |                 | Input          | 383     |
| В3      | PE5/TIA05       | Output enabled | 382     |
|         |                 | Output         | 381     |
|         |                 | Input          | 380     |
| C4      | PE4/TIA04       | Output enabled | 379     |
|         |                 | Output         | 378     |
|         |                 | Input          | 377     |

| Pin No. | Pin Name                | Input/Output   | Bit No. |
|---------|-------------------------|----------------|---------|
| C5      | PE3/TIA03               | Output enabled | 376     |
|         |                         | Output         | 375     |
|         |                         | Input          | 374     |
| C6      | PE2/TIA02/TIOC43/TIOC30 | Output enabled | 373     |
|         |                         | Output         | 372     |
|         |                         | Input          | 371     |
| D5      | PE1/TIA01/TIOC42/TIOC40 | Output enabled | 370     |
|         |                         | Output         | 369     |
|         |                         | Input          | 368     |
| D4      | PE0/TIA00               | Output enabled | 367     |
|         |                         | Output         | 366     |
|         |                         | Input          | 365     |
| C3      | PH5/TIF5                | Output enabled | 364     |
|         |                         | Output         | 363     |
|         |                         | <u> </u>       | 362     |
| D3      | PH4/TIF4                | Output enabled | 361     |
|         |                         | Output 3       | 360     |
|         |                         | Input          | 359     |
| B2      | PH3/TIF3                | Output enabled | 358     |
|         |                         | Output         | 357     |
|         |                         | Input          | 356     |
| F4      | PH2/TIF2A               | Output enabled | 355     |
|         |                         | Output         | 354     |
|         |                         | Input          | 353     |
| E3      | PH1/ADTRG_B/TIF1A       | Output enabled | 352     |
|         |                         | Output         | 351     |
|         |                         | Input          | 350     |
| B1      | PH0/ADTRG_A/TIF0A       | Output enabled | 349     |
|         |                         | Output         | 348     |
|         |                         | Input          | 347     |

| Pin No. | Pin Name         | Input/Output   | Bit No. |
|---------|------------------|----------------|---------|
| C2      | PK11/MISOC       | Output enabled | 346     |
|         |                  | Output         | 345     |
|         |                  | Input          | 344     |
| D2      | PK10/MOSIC       | Output enabled | 343     |
|         |                  | Output         | 342     |
|         |                  | Input          | 341     |
| F3      | PK9/RSPCKC       | Output enabled | 340     |
|         |                  | Output         | 339     |
|         |                  | Input          | 338     |
| E2      | PK8/RxD_E        | Output enabled | 337     |
|         |                  | Output         | 336     |
|         |                  | Input          | 335     |
| C1      | PK7/TxD_E        | Output enabled | 334     |
|         |                  | Output         | 333     |
|         |                  | Input          | 332     |
| D1      | PK6/SCK_E        | Output enabled | 331     |
|         |                  | Output         | 330     |
|         |                  | Input          | 329     |
| F2      | PK5/RxD_D/MISOB  | Output enabled | 328     |
|         |                  | Output         | 327     |
|         |                  | Input          | 326     |
| G3      | PK4/TxD_D/MOSIB  | Output enabled | 325     |
|         |                  | Output         | 324     |
|         |                  | Input          | 323     |
| E1      | PK3/SCK_D/RSPCKB | Output enabled | 322     |
|         |                  | Output         | 321     |
|         |                  | Input          | 320     |
| F1      | PK2/RxD_C/MISOA  | Output enabled | 319     |
|         |                  | Output         | 318     |
|         |                  | Input          | 317     |

| Pin No. | Pin Name                | Input/Output   | Bit No. |
|---------|-------------------------|----------------|---------|
| H4      | PK1/TxD_C/MOSIA         | Output enabled | 316     |
|         |                         | Output         | 315     |
|         |                         | Input          | 314     |
| G2      | PK0/SCK_C/RSPCKA/UBCTRG | Output enabled | 313     |
|         |                         | Output         | 312     |
|         |                         | Input          | 311     |
| G1      | PG0/TOD00A/SSLA0        | Output enabled | 310     |
|         |                         | Output         | 309     |
|         |                         | Input          | 308     |
| НЗ      | PG1/TOD01A/SSLA1        | Output enabled | 307     |
|         |                         | Output         | 306     |
|         |                         | Input          | 305     |
| H2      | PG2/TOD02A/SSLA2        | Output enabled | 304     |
|         |                         | Output         | 303     |
|         |                         | Input          | 302     |
| H1      | PG3/TOD03A/SSLA3        | Output enabled | 301     |
|         |                         | Output         | 300     |
|         |                         | Input          | 299     |
| J1      | PG4/TOD10A/SSLA4/SSLB3  | Output enabled | 298     |
|         |                         | Output         | 297     |
|         |                         | Input          | 296     |
| J3      | PG5/TOD11A/SSLA5/SSLC3  | Output enabled | 295     |
|         |                         | Output         | 294     |
|         |                         | Input          | 293     |
| J2      | PG6/TOD12A/SSLB0        | Output enabled | 292     |
|         |                         | Output         | 291     |
|         |                         | Input          | 290     |
| K4      | PG7/TOD13A/SSLB1        | Output enabled | 289     |
|         |                         | Output         | 288     |
|         |                         | Input          | 287     |

| Pin No. | Pin Name                   | Input/Output   | Bit No. |
|---------|----------------------------|----------------|---------|
| L1      | PG8/TOD20A/SSLB2/TIF6      | Output enabled | 286     |
|         |                            | Output         | 285     |
|         |                            | Input          | 284     |
| K3      | PG9/TOD21A/SSLC0/TIF7      | Output enabled | 283     |
|         |                            | Output         | 282     |
|         |                            | Input          | 281     |
| L2      | PG10/TOD22A/SSLC1/TIF8     | Output enabled | 280     |
|         |                            | Output         | 279     |
|         |                            | Input          | 278     |
| L3      | PG11/TOD23A/SSLC2/TIF9     | Output enabled | 277     |
|         |                            | Output         | 276     |
|         |                            | Input          | 275     |
| M1      | PG12/TOD30A/SSLA4/TIF10    | Output enabled | 274     |
|         |                            | Output         | 273     |
|         |                            | Input          | 272     |
| M4      | PG13/TOD31A/SSLA5/TIF11    | Output enabled | 271     |
|         |                            | Output         | 270     |
|         |                            | Input          | 269     |
| M2      | PG14/TOD32A/SSLA6/TIF12    | Output enabled | 268     |
|         |                            | Output         | 267     |
|         |                            | Input          | 266     |
| N1      | PG15/TOD33A/SSLA7/TIF13    | Output enabled | 265     |
|         |                            | Output         | 264     |
|         |                            | Input          | 263     |
| P1      | PL8/TOE33                  | Output enabled | 262     |
|         |                            | Output         | 261     |
|         |                            | Input          | 260     |
| M3      | PL7/TOE32/ <del>IRQ7</del> | Output enabled | 259     |
|         |                            | Output         | 258     |
|         |                            | Input          | 257     |

| Pin No. | Pin Name           | Input/Output   | Bit No. |
|---------|--------------------|----------------|---------|
| N2      | PL6/TOE31/IRQ6     | Output enabled | 256     |
|         |                    | Output         | 255     |
|         |                    | Input          | 254     |
| R1      | PL5/TOE30/IRQ5     | Output enabled | 253     |
|         |                    | Output         | 252     |
|         |                    | Input          | 251     |
| P2      | PL4/TOE23/IRQ4     | Output enabled | 250     |
|         |                    | Output         | 249     |
|         |                    | Input          | 248     |
| R2      | PL3/TOE22/IRQ3     | Output enabled | 247     |
|         |                    | Output         | 246     |
|         |                    | Input          | 245     |
| N3      | PL2/TOE21/IRQ2     | Output enabled | 244     |
|         |                    | Output         | 243     |
|         |                    | Input          | 242     |
| P4      | PL1/TOE20/IRQ1/POD | Output enabled | 241     |
|         |                    | Output         | 240     |
|         |                    | Input          | 239     |
| P3      | PL0/ĪRQ0           | Output enabled | 238     |
|         |                    | Output         | 237     |
|         |                    | Input          | 236     |
| V11     | PF0/TOD00B/TIF6    | Output enabled | 235     |
|         |                    | Output         | 234     |
|         |                    | Input          | 233     |
| Y13     | PF1/TOD01B/TIF7    | Output enabled | 232     |
|         |                    | Output         | 231     |
|         |                    | Input          | 230     |
| Y14     | PF2/TOD02B/TIF8    | Output enabled | 229     |
|         |                    | Output         | 228     |
|         |                    | Input          | 227     |

| Pin No. | Pin Name          | Input/Output   | Bit No. |
|---------|-------------------|----------------|---------|
| W13     | PF3/TOD03B/TIF9   | Output enabled | 226     |
|         |                   | Output         | 225     |
|         |                   | Input          | 224     |
| V12     | PF4/TOD10B/TIF10  | Output enabled | 223     |
|         |                   | Output         | 222     |
|         |                   | Input          | 221     |
| Y15     | PF5/TOD11B/TIF11  | Output enabled | 220     |
|         |                   | Output         | 219     |
|         |                   | Input          | 218     |
| Y16     | PF6/TOD12B/TIF12  | Output enabled | 217     |
|         |                   | Output         | 216     |
|         |                   | Input          | 215     |
| U13     | PF7/TOD13B/TIF13  | Output enabled | 214     |
|         |                   | Output         | 213     |
|         |                   | Input          | 212     |
| W14     | PF8/TOD20B/TIF14  | Output enabled | 211     |
|         |                   | Output         | 210     |
|         |                   | Input          | 209     |
| Y17     | PF9/TOD21B/TIF15  | Output enabled | 208     |
|         |                   | Output         | 207     |
|         |                   | Input          | 206     |
| V13     | PF10/TOD22B/TIF16 | Output enabled | 205     |
|         |                   | Output         | 204     |
|         |                   | Input          | 203     |
| Y18     | PF11/TOD23B/TIF17 | Output enabled | 202     |
|         |                   | Output         | 201     |
|         |                   | Input          | 200     |
| V14     | PF12/TOD30B/TIF18 | Output enabled | 199     |
|         |                   | Output         | 198     |
|         |                   | Input          | 197     |

| in No. | Pin Name                              | Input/Output   | Bit No. |
|--------|---------------------------------------|----------------|---------|
| W15    | PF13/TOD31B/TIF19                     | Output enabled | 196     |
|        |                                       | Output         | 195     |
|        |                                       | Input          | 194     |
| Y19    | PF14/TOD32B/CTx_B/TxD_A               | Output enabled | 193     |
|        |                                       | Output         | 192     |
|        |                                       | Input          | 191     |
| W16    | PF15/TOD33B/CRx_B/RxD_A               | Output enabled | 190     |
|        |                                       | Output         | 189     |
|        |                                       | Input          | 188     |
| W17    | PJ9/RxD_B                             | Output enabled | 187     |
|        |                                       | Output         | 186     |
|        |                                       | Input          | 185     |
| W18    | PJ8/TxD_B                             | Output enabled | 184     |
|        |                                       | Output         | 183     |
|        |                                       | Input          | 182     |
| U15    | PJ7/SCK_B/ADEND_A/TIJ1                | Output enabled | 181     |
|        |                                       | Output         | 180     |
|        |                                       | Input          | 179     |
| W19    | PJ6/RxD_A                             | Output enabled | 178     |
|        |                                       | Output         | 177     |
|        |                                       | Input          | 176     |
| V15    | PJ5/TxD_A                             | Output enabled | 175     |
|        |                                       | Output         | 174     |
|        |                                       | Input          | 173     |
| T17    | PJ4/SCK_A/ADEND_B/TIJ0                | Output enabled | 172     |
|        |                                       | Output         | 171     |
|        |                                       | Input          | 170     |
| U17    | PJ3/RxD_A/CRx_C/CRx_A&CRx_<br>B&CRx_C | Output enabled | 169     |
|        |                                       | Output         | 168     |
|        |                                       | Input          | 167     |

| Pin No. | Pin Name                              | Input/Output   | Bit No. |
|---------|---------------------------------------|----------------|---------|
| V16     | PJ2/TxD_A/CTx_C/CTx_A&CTx_<br>B&CTx_C | Output enabled | 166     |
|         |                                       | Output         | 165     |
|         |                                       | Input          | 164     |
| V17     | PJ1/RxD_A/CRx_A/CRx_A&CRx_B           | Output enabled | 163     |
|         |                                       | Output         | 162     |
|         |                                       | Input          | 161     |
| V18     | PJ0/TxD_A/CTx_A/CTx_A&CTx_B           | Output enabled | 160     |
|         |                                       | Output         | 159     |
|         |                                       | Input          | 158     |
| T18     | PC15/D15                              | Output enabled | 157     |
|         |                                       | Output         | 156     |
|         |                                       | Input          | 155     |
| U18     | PC14/D14                              | Output enabled | 154     |
|         |                                       | Output         | 153     |
|         |                                       | Input          | 152     |
| V19     | PC13/D13                              | Output enabled | 151     |
|         |                                       | Output         | 150     |
|         |                                       | Input          | 149     |
| W20     | PC12/D12                              | Output enabled | 148     |
|         |                                       | Output         | 147     |
|         |                                       | Input          | 146     |
| V20     | PC11/D11                              | Output enabled | 145     |
|         |                                       | Output         | 144     |
|         |                                       | Input          | 143     |
| R18     | PC10/D10                              | Output enabled | 142     |
|         |                                       | Output         | 141     |
|         |                                       | Input          | 140     |
| U19     | PC9/D9                                | Output enabled | 139     |
|         |                                       | Output         | 138     |
|         |                                       | Input          | 137     |

| Pin No. | Pin Name   | Input/Output   | Bit No. |
|---------|------------|----------------|---------|
| U20     | PC8/D8     | Output enabled | 136     |
|         |            | Output         | 135     |
|         |            | Input          | 134     |
| P18     | PC7/D7     | Output enabled | 133     |
|         |            | Output         | 132     |
|         |            | Input          | 131     |
| T19     | PC6/D6     | Output enabled | 130     |
|         |            | Output         | 129     |
|         |            | Input          | 128     |
| R19     | PC5/D5     | Output enabled | 127     |
|         |            | Output         | 126     |
|         |            | Input          | 125     |
| T20     | PC4/D4     | Output enabled | 124     |
|         |            | Output         | 123     |
|         |            | Input          | 122     |
| N17     | PC3/D3     | Output enabled | 121     |
|         |            | Output         | 120     |
|         |            | Input          | 119     |
| P19     | PC2/D2     | Output enabled | 118     |
|         |            | Output         | 117     |
|         |            | Input          | 116     |
| R20     | PC1/D1     | Output enabled | 115     |
|         |            | Output         | 114     |
|         |            | Input          | 113     |
| N18     | PC0/D0     | Output enabled | 112     |
|         |            | Output         | 111     |
|         |            | Input          | 110     |
| P20     | PB14/RD/WR | Output enabled | 109     |
|         |            | Output         | 108     |
|         |            | Input          | 107     |

| Pin No. | Pin Name           | Input/Output   | Bit No. |
|---------|--------------------|----------------|---------|
| N19     | PB13/CS3/RSPCKB    | Output enabled | 106     |
|         |                    | Output         | 105     |
|         |                    | Input          | 104     |
| N20     | PB12/CS2/RSPCKA    | Output enabled | 103     |
|         |                    | Output         | 102     |
|         |                    | Input          | 101     |
| M20     | PB11/CS1/TOE21     | Output enabled | 100     |
|         |                    | Output         | 99      |
|         |                    | Input          | 98      |
| M18     | PB10/CS0           | Output enabled | 97      |
|         |                    | Output         | 96      |
|         |                    | Input          | 95      |
| M19     | PB9/RD             | Output enabled | 94      |
|         |                    | Output         | 93      |
|         |                    | Input          | 92      |
| K20     | PB8/WAIT/TOE20     | Output enabled | 91      |
|         |                    | Output         | 90      |
|         |                    | Input          | 89      |
| L17     | PB7/WE1            | Output enabled | 88      |
|         |                    | Output         | 87      |
|         |                    | Input          | 86      |
| K19     | PB6/WE0            | Output enabled | 85      |
|         |                    | Output         | 84      |
|         |                    | Input          | 83      |
| J20     | PB5/A21/CRx_B/TIF7 | Output enabled | 82      |
|         |                    | Output         | 81      |
|         |                    | Input          | 80      |
| H20     | PB4/A20/CTx_B/TIF6 | Output enabled | 79      |
|         |                    | Output         | 78      |
|         |                    | Input          | 77      |

| Pin No. | Pin Name      | Input/Output   | Bit No. |
|---------|---------------|----------------|---------|
| L18     | PB3/A19/MISOB | Output enabled | 76      |
|         |               | Output         | 75      |
|         |               | Input          | 74      |
| J19     | PB2/A18/MOSIB | Output enabled | 73      |
|         |               | Output         | 72      |
|         |               | Input          | 71      |
| K18     | PB1/A17/MISOA | Output enabled | 70      |
|         |               | Output         | 69      |
|         |               | Input          | 68      |
| G20     | PB0/A16/MOSIA | Output enabled | 67      |
|         |               | Output         | 66      |
|         |               | Input          | 65      |
| F20     | PA15/A15      | Output enabled | 64      |
|         |               | Output         | 63      |
|         |               | Input          | 62      |
| H19     | PA14/A14      | Output enabled | 61      |
|         |               | Output         | 60      |
|         |               | Input          | 59      |
| E20     | PA13/A13      | Output enabled | 58      |
|         |               | Output         | 57      |
|         |               | Input          | 56      |
| D20     | PA12/A12      | Output enabled | 55      |
|         |               | Output         | 54      |
|         |               | Input          | 53      |
| G19     | PA11/A11      | Output enabled | 52      |
|         |               | Output         | 51      |
|         |               | Input          | 50      |
| J18     | PA10/A10      | Output enabled | 49      |
|         |               | Output         | 48      |
|         |               | Input          | 47      |

| Pin No. | Pin Name | Input/Output   | Bit No. |
|---------|----------|----------------|---------|
| C20     | PA9/A9   | Output enabled | 46      |
|         |          | Output         | 45      |
|         |          | Input          | 44      |
| H18     | PA8/A8   | Output enabled | 43      |
|         |          | Output         | 42      |
|         |          | Input          | 41      |
| B20     | PA7/A7   | Output enabled | 40      |
|         |          | Output         | 39      |
|         |          | Input          | 38      |
| F19     | PA6/A6   | Output enabled | 37      |
|         |          | Output         | 36      |
|         |          | Input          | 35      |
| G17     | PA5/A5   | Output enabled | 34      |
|         |          | Output         | 33      |
|         |          | Input          | 32      |
| D19     | PA4/A4   | Output enabled | 31      |
|         |          | Output         | 30      |
|         |          | Input          | 29      |
| E19     | PA3/A3   | Output enabled | 28      |
|         |          | Output         | 27      |
|         |          | Input          | 26      |
| C19     | PA2/A2   | Output enabled | 25      |
|         |          | Output         | 24      |
|         |          | Input          | 23      |
| B19     | PA1/A1   | Output enabled | 22      |
|         |          | Output         | 21      |
|         |          | Input          | 20      |
| C18     | PA0/A0   | Output enabled | 19      |
|         |          | Output         | 18      |
|         |          | Input          | 17      |

| AUDSYNC | Output                                   | 16  |
|---------|--|---|
|         | Output enabled                           | 15  |
|         | Input                                    | 14  |
| AUDATA0 | Output                                   | 13  |
|         | Output enabled                           | 12  |
|         | Input                                    | 11  |
| AUDATA1 | Output                                   | 10  |
|         | Output enabled                           | 9   |
|         | Input                                    | 8   |
| AUDATA2 | Output                                   | 7   |
|         | Output enabled                           | 6   |
|         | Input                                    | 5   |
| AUDATA3 | Output                                   | 4   |
|         | Output enabled                           | 3   |
|         | Input                                    | 2   |
| AUDMD   | Input                                    | 1   |
| AUDRST  | Input                                    | 0   |
| To TDO  |  |   |
|         | AUDATA1  AUDATA2  AUDATA3  AUDMD  AUDRST | Output enabled   Input   Output   Output   Output enabled   Input   Input   AUDMD   Input   Input |

# 20.4 Operations

#### 20.4.1 TAP Controller

Figure 20.2 shows the internal state transition of the TAP controller.

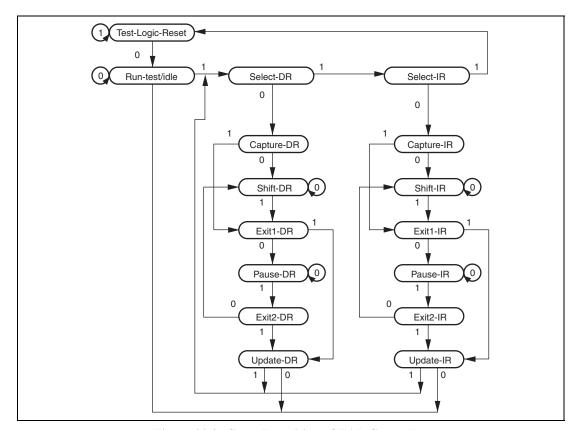


Figure 20.2 State Transition of TAP Controller

Note: A transition occurs depending on the TMS value at the rising edge of the TCK signal. The TDI value is sampled at the rising edge of the TCK signal and is shifted at the falling edge of the TCK signal. The TDO pin is always in the high-impedance state except in the Shift-DR and Shift-IR states. When the TRST pin is asserted, a transition to the Test-Logic-Reset state occurs asynchronously with the TCK signal.

SH72546R Section 20 JTAG Interface

### 20.4.2 Supported Commands

#### (1) BYPASS

The BYPASS command is an indispensable standard command for operating SDBPR. This command shortens the shift path thus transferring serial data for other LSIs on a printed-circuit board at a higher speed. While this command is being executed, the test circuit has no effect on the system circuit.

#### (2) SAMPLE/PRELOAD

The SAMPLE/PRELOAD command inputs data to SDBSR from the internal circuits of this LSI, outputs the data via the scan path, and loads data to the scan path. While this command is being executed, the values of the input pins of this LSI are sent directly to the internal circuits of this LSI, whereas the values of the internal circuits are output directly from the output pins. Executing this command has no effect on the system circuit.

The SAMPLE operation latches the snapshots of the values transferred from the input pins to the internal circuits and the values transferred from the internal circuits to the output pins and reads out the snapshots via the scan path. Here, the snapshots are latched without disturbing the operations of this LSI.

The PRELOAD operation sets the initial value to the parallel output latch circuit of the SDBSR via the scan path, prior to the EXTEST command. If the EXTEST command is executed without PRELOAD operation, undefined data is output from the output pins (with the EXTEST command, the values are always output to the output pins from the parallel output latch circuit) until the first scan sequence is complete (until data transfer to the output latch circuit is complete).

#### (3) EXTEST

The EXTEST command tests external circuits when this LSI is mounted on a printed-circuit board. When this command is executed, the output pins are used to output the test data, which has been set using the SAMPLE/PRELOAD command, to the printed-circuit board from the SDBSR, whereas the input pins are used to latch the test result to SDBSR from the printed-circuit board. When the EXTEST command is used N times for a test, the test data for the Nth command is scanned in when the test data for the (N-1)th command is scanned out.

The data that has been loaded on SDBSR of the output pins in the Capture-DR state of this command is not used to test the external circuits (it is shifted and replaced).

Section 20 JTAG Interface SH72546R

#### (4) CLAMP

The CLAMP command allows the output pins to output the value from SDBSR, which has been set using the SAMPLE/PRELOAD command. While the CLAMP command is being selected, SDBSR retains the previous state regardless of the TAP controller state.

SDBPR is connected between the TDI and TDO pins, thus allowing the same operation as that when the BYPASS command is selected.

#### (5) HIGHZ

The HIGHZ command drives all the output pins to the high-impedance state. While the HIGHZ command is being selected, SDBSR retains the previous state regardless of the TAP controller state.

SDBPR is connected between the TDI and TDO pins, thus allowing the same operation as that when the BYPASS command is selected.

#### (6) IDCODE

The IDCODE command sets the JTAG interface pins to IDCODE mode prescribed in the JTAG. Specifically, the pins are set to IDCODE mode when the JTAG interface is initialized, that is, when the TRST pin is asserted or the TAP controller is placed in the Test-Logic-Reset state.

#### 20.4.3 Notes

The JTAG interface of this LSI has the following restrictions.

- Clock-related signals (EXTAL and XTAL) cannot be boundary-scanned.
- $\bullet$  Reset-related signals ( $\overline{RES}$  and  $\overline{HSTBY})$  cannot be boundary-scanned.
- JTAG interface-related signals (TCK, TDI, TDO, TMS, and TRST) cannot be boundary-scanned.
- MD0, MD1, MD2, MD4, ASEMD, AUDCK, and CK cannot be boundary-scanned.
- When using the JTAG interface, pins ASEMD, MD0, MD1, MD2, and MD4 must be 0 and pins RES and HSTBY must be 1.
- ADC-related pins (AN0 to AN27 and AN40 to AN48) cannot be boundary-scanned.
- Fix the RES pin to the low level while the EXTEST, CLAMP, or HIGHZ command is being set.
- Selecting HIGHZ command for the following pins is invalid, as they are set to be pulled up/down.

Always invalid: AUDMD, AUDSYNC, AUDATA, and AUDRST

Invalid only when being set to be pulled up/down via the I/O port (refer to section 23, I/O Ports):

PB1, PB3, PF15, PJ1, PJ3, PJ6, PJ9, PK2, PK5, PK8, and PK11

Section 20 JTAG Interface SH72546R

## 20.5 Usage Notes

 Once a command is set, it is not updated until a different command is issued. When using the same command repeatedly, temporarily set a command that has no effect on the LSI's operation (e.g., BYPASS command) and set the desired command again.

- 2. No commands are accepted in hardware standby mode.
- 3. Whether or not the JTAG interface is in use, be sure to hold the TRST signal at the low level when resetting the chip. Specifically, hold the TRST signal low for 20 TCK clock cycles. For details, see section 32, Electrical Characteristics.
- 4. Up to 2 MHz frequency can be input to the TCK. Normal operation cannot be guaranteed if the frequency exceeds 2 MHz. For details, see section 32, Electrical Characteristics.
- 5. If data is transferred serially to the register that connects between the TDI and TDO pins and the data has a larger number of bits than the number of bits in that register, serial data bits that are in excess of the number of bits in the register are output from the TDO pin. In this case, such excess bits are from the data that has been input via the TDI pin.
- 6. If the serial transfer sequence failed, be sure to reset the JTAG interface using the TRST pin. Here, repeat the data transfer in question from the beginning regardless of the point at which the transfer failed.
- 7. The TDO pin starts outputting data at the falling edge of the TCK signal.
- 8. For easy debugging, design the TRST signal line pattern on the board so that it can be easily cut.
- 9. Do not halt this module by using the module standby function during boundary-scan. Instead, put the module into sleep mode immediately after a reset, then perform boundary scan. Refer to section 29, Power-Down Modes for details on the module standby function and the sleep mode.

# Section 21 Advanced User Debugger II (AUD-II)

AUD-II provides functions for debugging the user program with this LSI mounted on the board. Using AUD-II facilitates configuration of a simple emulation system having functions such as acquisition of AUD tracing data and monitoring or tuning the contents of the on-chip RAM.

#### 21.1 Features

AUD-II has the following two modes which can be switched by the setting of the AUDMD pin.

- AUD tracing mode
- RAM monitoring mode

### (1) AUD Tracing Mode

- Eight input/output pins
- Branch tracing function

Tracing both branch source and branch destination or tracing either branch source or branch destination can be selected

Window data tracing function

Window A and window B are available. When memory in a window area is accessed by the CPU or DMAC, the address and data are traced. Data lengths of 8, 16, and 32 bits are supported. M bus and I bus can be traced.

- Only the lower bits of an address, which are the difference from the previously accessed address, are output for efficient use of the AUDATA pins.
- Eight-stage FIFO
- Full tracing function

All traced data is output even when the amount of information to be output exceeds the amount of traced information as making the CPU stall.

• Realtime tracing function

The amount of traced information is reduced so that the CPU does not stall.

• AUD output clock selection

The frequency of the AUD output clock is 1/4, 1/8, or 1/10 of that of the internal clock ( $\phi$ ).

## (2) RAM Monitor Mode

- All memory-mapped modules connected to internal or external buses can be read from or written to.
- When an address is input on the AUDATA pins, data at the address is output.
- When an address and data are input on the AUDATA pins, the data is written to the address.

Note: In this section, AUDATA[3:0] denotes AUDATA 3 to AUDATA 0.

Figure 21.1 is a block diagram of AUD-II.

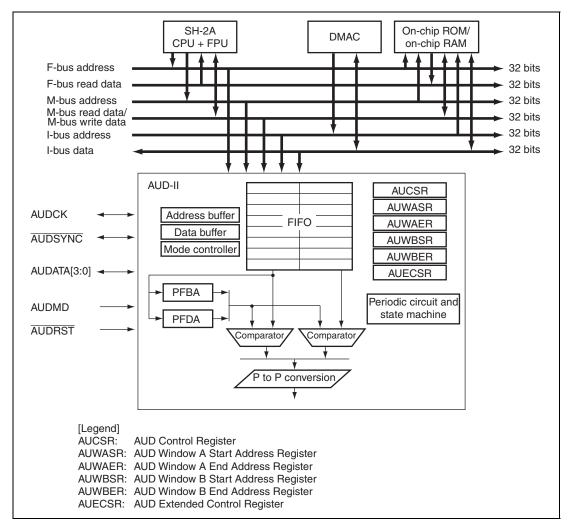


Figure 21.1 Block Diagram of AUD-II

## 21.2 Input/Output Pins

Table 21.1 shows the pin configuration of AUD-II.

**Table 21.1 Pin Configuration** 

|           |             | Descr  | iption  |
|-----------|-------------|--|---|
| Pin Name  | Symbol      | AUD Tracing Mode   | RAM Monitoring Mode   |
| AUD reset | AUDRST      | Input pin for an AUD-II reset signal                         | Input pin for an AUD-II reset signal                              |
| AUD sync  | AUDSYNC     | Output pin for the data start position identification signal | Input pin for the data start position identification signal       |
| AUD clock | AUDCK       | Output pin for the clock signal                              | Input pin for the external clock signal                           |
| AUD mode  | AUDMD       | Input pin for the mode selecting signal (low)                | Input pin for the mode selecting signal (high)                    |
| AUD data  | AUDATA[3:0] | Output pins for traced data and addresses                    | Input pins for monitoring addresses or input/output pins for data |

## 21.2.1 Commonly Used Pins

**Table 21.2 Commonly Used Pins** 

| Symbol | Description   |
|--------|---|
| AUDMD  | The following mode can be selected by the level on this pin.  |
|        | Low: AUD tracing mode   |
|        | High: RAM monitoring mode   |
|        | The level on this pin must be changed while the $\overline{\text{AUDRST}}$ signal is low. When this pin is not connected, it is pulled up internally.   |
| AUDRST | While a low level is input on this pin, AUD-II is in the reset state. Buffers and internal states of AUD-II are initialized. When the level on the AUDMD pin is driven high after it has been stable, AUD-II starts in the selected mode. When this pin is not connected, it is pulled down internally. |

## 21.2.2 Pin Descriptions in AUD Tracing Mode

## Table 21.3 Pin Descriptions in AUD Tracing Mode

| Symbol      | Description  |  |  |  |  |  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|--|--|--|--|--|
| AUDCK       | Outputs the clock signal to synchronize the AUDATA signals.  |  |  |  |  |  |  |  |  |  |  |
|             | The frequency of this clock signal is 1/4, 1/8, or 1/10 of the frequency of the internal clock $(\phi).$     |  |  |  |  |  |  |  |  |  |  |
| AUDSYNC     | Indicates the AUD bus command being output.  |  |  |  |  |  |  |  |  |  |  |
|             | Indicates that AUD bus command 1 (CMD1) is being output  |  |  |  |  |  |  |  |  |  |  |
|             | <ol><li>Indicates that AUD bus command 2 (CMD2), addresses, and data are being<br/>output</li></ol>          |  |  |  |  |  |  |  |  |  |  |
|             | Regardless of the data, a level of 0 is always output in the next cycle in which the LOST command is output. |  |  |  |  |  |  |  |  |  |  |
| AUDATA[3:0] | These pins output the following information at a specified timing.   |  |  |  |  |  |  |  |  |  |  |
|             | AUD bus command  |  |  |  |  |  |  |  |  |  |  |
|             | Branch source and destination addresses  |  |  |  |  |  |  |  |  |  |  |
|             | Address and data for window tracing  |  |  |  |  |  |  |  |  |  |  |

## 21.2.3 Pin Descriptions in RAM Monitoring Mode

## **Table 21.4 Pin Descriptions in RAM Monitoring Mode**

| Symbol      | Description   |
|-------------|---|
| AUDCK       | Input the external clock signal to be used for debugging.   |
|             | The input frequency must be less than the frequency from the EXTAL pin. When this pin is not connected, it is pulled up internally.   |
| AUDSYNC     | Input the signal which indicates that the AUD bus command is valid.   |
|             | 1: While this signal is driven high, read data is output  |
|             | 0: While this signal is driven low, a write address, write data, and a DIR command can be input and the ready code is output  |
|             | When this pin is not connected, it is pulled up internally.   |
|             | Note: Assert this signal after a required command has been ready on the AUDATA pins. For details, see section 21.4, RAM Monitoring Mode.  |
| AUDATA[3:0] | These pins output the following information at a specified timing.  |
|             | AUD bus command   |
|             | Addresses   |
|             | • Data  |
|             | Input a command on these pins. Then, the information is output after the ready code is output. When the $\overline{\text{AUDSYNC}}$ signal is negated, the information starts to be output. |
|             | For details, see section 21.4, RAM Monitoring Mode. When this pin is not connected, it is pulled up internally.   |

## 21.3 AUD Tracing Mode

## 21.3.1 Register Descriptions

For controlling the various functions of AUD-II, AUD-II has registers AUCSR and AUECSR. For setting windows, registers AUWASR, AUWBSR, AUWAER, and AUWBER are available. Table 21.5 shows details of the register configuration.

**Table 21.5 Register Configuration** 

| Register Name                       | Abbreviation | R/W | Initial<br>Value | Address    | Access Size |
|-------------------------------------|--------------|-----|------------------|------------|-------------|
| AUD control register                | AUCSR        | R/W | H'0000           | H'FFFC0C00 | 8, 16       |
| AUD window A start address register | AUWASR       | R/W | Undefined        | H'FFFC0C04 | 8, 16, 32   |
| AUD window A end address register   | AUWAER       | R/W | Undefined        | H'FFFC0C08 | 8, 16, 32   |
| AUD window B start address register | AUWBSR       | R/W | Undefined        | H'FFFC0C0C | 8, 16, 32   |
| AUD window B end address register   | AUWBER       | R/W | Undefined        | H'FFFC0C10 | 8, 16, 32   |
| AUD extended control register       | AUECSR       | R/W | H'0000           | H'FFFC0C14 | 8, 16       |

### 21.3.2 AUD Control Register (AUCSR)

AUCSR is a 16-bit readable/writable register.

| Bit:           | 15       | 14       | 13       | 12       | 11       | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|----------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                | CLK      | [1:0]    | BW[      | 1:0]     | OC[1:0]  |          | BR[1:0]  |          | WA[1:0]  |          | WB[1:0]  |          | -        | TM       | -        | EN       |
| Initial value: | 0<br>R/W |

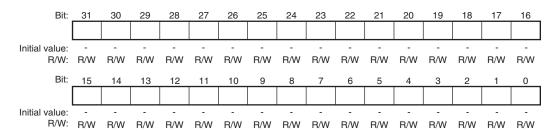
| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 15, 14 | CLK[1:0] | 00               | R/W | AUD Clock Select   |
|        |          |                  |     | These bits set the ratio of the frequency of the clock signal output on the AUDCK pin to the internal clock (φ). For restrictions on the AUDCK clock ratio, see a description of Rules on AUDCK, under section 21.3.9 (2), Rules on AUDCK.   |
|        |          |                  |     | 00: A ratio of 1 to 8  |
|        |          |                  |     | 01: A ratio of 1 to 4  |
|        |          |                  |     | 10: A ratio of 1 to 10   |
|        |          |                  |     | 11: Reserved   |
|        |          |                  |     | Note: CLK[1:0] must be changed only when the EN bit in AUCSR is 0.   |
| 13, 12 | BW[1:0]  | 00               | R/W | AUD Output Bus Width   |
|        |          |                  |     | These bits specify the bus width for the AUDATA pins. These bits are provided for future use. So they are not yet actually implemented. Similar to other reserved bits, when writing to either of these bits, only 0 should be written to. Both of these bits are read as 0.   |
|        |          |                  |     | 00: 4-bit mode. Trace information is output on pins AUDATA[3:0].   |
|        |          |                  |     | 01: Reserved   |
|        |          |                  |     | 10: Reserved   |
|        |          |                  |     | 11: Reserved   |
| 11, 10 | OC[1:0]  | 00               | R/W | Output Counter Mode  |
|        |          |                  |     | Traced information to be output, such as information for branch tracing, includes only the lower address bits. However, all the 32 bits of the address can be output at specified intervals. The interval can be set in these bits. When these bits are set to B'11, only the difference from the previous traced address is output on the AUDATA pins other than the first timer after a reset. |
|        |          |                  |     | When these bits are cleared to B'00, all the address bits are output once per 128 times.   |
|        |          |                  |     | 00: All the address bits are output once per 128 times   |
|        |          |                  |     | 01: Reserved   |
|        |          |                  |     | 10: Reserved   |
|        |          |                  |     | 11: The lower bits changed from the previous address   |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 9, 8 | BR[1:0]  | 00               | R/W | Branch Tracing function  |
|      |          |                  |     | AUD-II traces the branch destination and/or source addresses according to the setting of these bits.   |
|      |          |                  |     | 00: Branch tracing disabled  |
|      |          |                  |     | 01: Branch tracing enabled; branch source and destination are output   |
|      |          |                  |     | <ol> <li>Branch tracing enabled; only branch source is<br/>output</li> </ol>   |
|      |          |                  |     | <ol> <li>Branch tracing enabled; only branch destination is<br/>output</li> </ol>  |
|      |          |                  |     | The setting in these bits takes effect when data is output after data next to the current data has been output.  |
| 7, 6 | WA[1:0]  | 00               | R/W | Window A Data Tracing Function   |
|      |          |                  |     | AUD-II traces memory access to the area specified as window A according to the setting of these bits. As tracing conditions, either read access or write access, or both can be specified. |
|      |          |                  |     | Because multiple buses are in this LSI, the bus to be traced can be selected. For details, see section 21.3.7, AUD Extended Control Register (AUECSR).                                     |
|      |          |                  |     | 00: Window A data tracing function disabled  |
|      |          |                  |     | 01: Only write access is to be traced  |
|      |          |                  |     | 10: Only read access is to be traced   |
|      |          |                  |     | 11: Read and write accesses are to be traced   |
| 5, 4 | WB[1:0]  | 00               | R/W | Window B Data Tracing Function   |
|      |          |                  |     | AUD-II traces memory access to the area specified as window B according to the setting of these bits. As tracing conditions, either read access or write access, or both can be specified. |
|      |          |                  |     | Because multiple buses are in this LSI, the bus to be traced can be selected. For details, see section 21.3.7, AUD Extended Control Register (AUECSR).                                     |
|      |          |                  |     | 00: Window B data tracing function disabled  |
|      |          |                  |     | 01: Only write access is to be traced  |
|      |          |                  |     | 10: Only read access is to be traced   |
|      |          |                  |     | 11: Read and write accesses are to be traced   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 3   | _        | 0                | R/W | Reserved   |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 2   | TM       | 0                | R/W | Tracing mode   |
|     |          |                  |     | Specifies the operation of the CPU when the FIFO buffer for storing various trace information on AUD-II is full. |
|     |          |                  |     | Full tracing mode in which all traced events are<br>output   |
|     |          |                  |     | Realtime tracing mode in which realtime tracing information is output without stopping the CPU                   |
| 1   | _        | 0                | R/W | Reserved   |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 0   | EN       | 0                | R/W | AUD Trace Enable   |
|     |          |                  |     | Enables tracing function specified by bits BR[1:0], WA[1:0], and WB[1:0].  |
|     |          |                  |     | When this bit is 0, the tracing function specified by bits BR[1:0], WA[1:0], and WB[1:0] bits is not executed.   |
|     |          |                  |     | 0: AUD tracing function disabled   |
|     |          |                  |     | 1: AUD tracing function enabled  |

#### 21.3.3 AUD Window A Start Address Register (AUWASR)

AUWASR is a 32-bit readable/writable register that sets the logical address of the start address of window A to be traced. The end address of window A is specified by the AUD window A end address register (AUWAER). Window A is an area from the address specified by AUWASR to the address specified by AUWAER. AUWASR  $\leq$  window A  $\leq$  AUWAER.



#### 21.3.4 AUD Window A End Address Register (AUWAER)

AUWAER is a 32-bit readable/writable register that sets the logical address of the end address of window A to be traced. The start address of window A is specified by the AUD window A start address register (AUWASR).

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| R/W·           | R/W |

#### 21.3.5 AUD Window B Start Address Register (AUWBSR)

AUWBSR is a 32-bit readable/writable register that sets the logical address of the start address of window B to be traced. The end address is specified by the AUD window B end address register (AUWBER). Window B is an area from the address specified by AUWBSR to the address specified by AUWBER. AUWBSR  $\leq$  window B  $\leq$  AUWBER.

| Bit:           | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| R/W:           | R/W |
| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Initial value: | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   | -   |
| R/W:           | R/W |

### 21.3.6 AUD Window B End Address Register (AUWBER)

AUWBER is a 32-bit readable/writable register that sets the logical address of the end address of window B to be traced. The start address of window B is specified by the AUD window B start address register (AUWBSR).

| Bit:                   | 31       | 30       | 29       | 28       | 27       | 26       | 25       | 24       | 23       | 22       | 21       | 20       | 19       | 18       | 17       | 16       |
|------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | -<br>R/W |
| Bit:                   | 15       | 14       | 13       | 12       | 11       | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|                        |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | -<br>R/W |

## 21.3.7 AUD Extended Control Register (AUECSR)

AUECSR is a 16-bit readable/writable register.

| Bit:           | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8       | 7   | 6   | 5       | 4   | 3    | 2    | 1    | 0   |
|----------------|-----|-----|-----|-----|-----|-----|-----|---------|-----|-----|---------|-----|------|------|------|-----|
|                | -   | -   | -   | -   | -   | -   | W   | /A0B[2: | 0]  | V   | /B0B[2: | :0] | TREX | TRSB | TRGN | -   |
| Initial value: | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0       | 0   | 0   | 0       | 0   | 0    | 0    | 0    | 0   |
| R/W:           | R/W     | R/W | R/W | R/W     | R/W | R/W  | R/W  | R/W  | R/W |

| Bit     | Bit Name  | Initial<br>Value | R/W | Description  |
|---------|-----------|------------------|-----|--|
| 15 to10 | _         | All 0            | R/W | Reserved   |
|         |           |                  |     | These bits are always read as 0. The write value should always be 0.         |
| 9 to 7  | WA0B[2:0] | 000              | R/W | Window A Trace Bus Select  |
|         |           |                  |     | These bits specify the internal bus to be traced.                            |
|         |           |                  |     | 000: Reserved  |
|         |           |                  |     | 001: M bus   |
|         |           |                  |     | 010: Reserved  |
|         |           |                  |     | 011: Reserved  |
|         |           |                  |     | 100: I bus   |
|         |           |                  |     | 101: Reserved  |
|         |           |                  |     | 110: Reserved  |
|         |           |                  |     | 111: Reserved  |
|         |           |                  |     | Note that the WA0B[2:0] bits must be modified when the EN bit in AUCSR is 0. |

| Bit    | Bit Name  | Initial<br>Value | R/W | Description   |
|--------|-----------|------------------|-----|---|
| 6 to 4 | WB0B[2:0] | 000              | R/W | Window B Trace Bus Select   |
|        |           |                  |     | These bits specify the internal bus to be traced.   |
|        |           |                  |     | 000: Reserved   |
|        |           |                  |     | 001: M bus  |
|        |           |                  |     | 010: Reserved   |
|        |           |                  |     | 011: Reserved   |
|        |           |                  |     | 100: I bus  |
|        |           |                  |     | 101: Reserved   |
|        |           |                  |     | 110: Reserved   |
|        |           |                  |     | 111: Reserved   |
|        |           |                  |     | Note that the WB0B[2:0] bits must be modified when the EN bit in AUCSR is 0.  |
| 3      | TREX      | 0                | R/W | Exception Branch Trace Select   |
|        |           |                  |     | When the branch tracing function is enabled, this bit specifies whether or not an exception branch is to be traced. |
|        |           |                  |     | 0: Exception branches traced  |
|        |           |                  |     | 1: Exception branches not traced  |
| 2      | TRSB      | 0                | R/W | Subroutine Branch Trace Select  |
|        |           |                  |     | When the branch tracing function is enabled, this bit specifies whether or not a subroutine branch is to be traced. |
|        |           |                  |     | 0: Subroutine branches traced   |
|        |           |                  |     | 1: Subroutine branches not traced   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 1   | TRGN     | 0                | R/W | General Branch Trace Select  |
|     |          |                  |     | When the branch tracing function is enabled, this bit specifies whether or not a general branch is to be traced.   |
|     |          |                  |     | 0: General branches traced   |
|     |          |                  |     | 1: General branches not traced   |
|     |          |                  |     | Note: Branches are broken down into categories as follows:   |
|     |          |                  |     | Exception branch: general illegal instruction, slot illegal instruction, bank underflow, interrupt operation, TRAPA instruction, RTE instruction, UBC break (PC), and power-on reset |
|     |          |                  |     | Sub-routine branch: BSR, BSRF, JSR, JSR/N, RTS, RTS/N, and RTV/N instructions  |
|     |          |                  |     | General branch: BF, BT, BF/S, BT/S, BRA, BRAF, and JMP instructions  |
| 0   | _        | 0                | R/W | Reserved   |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.  |

### 21.3.8 Operation

To use the AUD tracing function, perform the following procedure in this order.

- 1. While the AUDRST signal is asserted, negate the AUDMD signal.
- 2. Negate the  $\overline{AUDRST}$  signal.
- 3. Set AUCSR, AUWASR, AUWAER, AUWBSR, AUWBER, and AUECSR.
- 4. Set the EN bit in AUCSR to 1.

AUD-II supports a branch tracing and a window data tracing functions. These functions can be executed independently. Further, for external output of the data traced by these functions, the user can select the realtime tracing mode or full tracing mode. Detailed operations of these AUD tracing modes are described below.

#### (1) AUD Bus Command

The information traced by AUD-II is output in a packet format on pins AUDATA[3:0] and AUDSYNC in synchronization with the AUDCK signal. The packet is comprised of a command part and a data part that contains 0 to 16 units of data. Packets are output continuously on pins AUDATA[3:0]. Normally, the command part is comprised of CMD1, which indicates the type of packet, and CMD2, which indicates the packet length of the data part. In addition, there is a special packet comprised solely of CMD1 that indicates the status of AUD-II. The AUDSYNC signal goes high when the bus is idle or when CMD1 is output on pins AUDATA[3:0]; and it goes low when CMD2, addresses, and data are output. Table 21.6 shows details of commands.

**Table 21.6 List of AUD Bus Commands** 

| Command | CMD1   | CMD2       | Descriptions   |
|---------|--------|------------|--|
| STDBY   | B'0000 | _          | This command indicates the standby status.   |
|         |        |            | There is no subsequent data part to be output.   |
| LOST    | B'0001 | _          | In realtime tracing mode, this command indicates that the data to be output has been lost.   |
|         |        |            | In full tracing mode, this command indicates that the FIFO for AUD-II is full or the CPU stops temporarily because multiple tracing occurs simultaneously.   |
| BGC     | B'0010 | (sda)(ssa) | Indicates that branch tracing information (general branching) is to be output. In the data part following this command, a branch destination address and a branch source address are output in the order on pins AUDATA[3:0].    |
|         |        |            | sda: Indicates the size of a branch destination address.  Disused when the tracing is only for the branch source.  |
|         |        |            | ssa: Indicates the size of a branch source address. Disused when the tracing is only for the branch destination.   |
|         |        |            | sda/ssa = B'00: The lower 4 bits of the address  |
|         |        |            | sda/ssa = B'01: The lower 8 bits of the address  |
|         |        |            | sda/ssa = B'10: The lower 16 bits of the address   |
|         |        |            | sda/ssa = B'11: All the 32-bits of address (full address)  |
| BSC     | B'0011 | (sda)(ssa) | Indicates that branch tracing information (subroutine branching) is to be output. In the data part following this command, a branch destination address and a branch source address are output in the order on pins AUDATA[3:0]. |
|         |        |            | sda: Indicates the size of a branch destination address.   |
|         |        |            | ssa: Indicates the size of a branch source address.  |
|         |        |            | sda/ssa = B'00: The lower 4 bits of the address  |
|         |        |            | sda/ssa = B'01: The lower 8 bits of the address  |
|         |        |            | sda/ssa = B'10: The lower 16 bits of the address   |
|         |        |            | sda/ssa = B'11: All the 32-bits of address (full address)  |

| Command | CMD1   | CMD2       | Descriptions  |
|---------|--------|------------|---|
| BEC     | B'0100 | (sda)(ssa) | Indicates that branch tracing information (exception branching) is to be output. In the data part following this command, a branch destination address and a branch source address are output in the order on pins AUDATA[3:0]. |
|         |        |            | sda: Indicates the size of a branch destination address.  |
|         |        |            | ssa: Indicates the size of a branch source address.   |
|         |        |            | sda/ssa = B'00: The lower 4 bits of the address   |
|         |        |            | sda/ssa = B'01: The lower 8 bits of the address   |
|         |        |            | sda/ssa = B'10: The lower 16 bits of the address  |
|         |        |            | sda/ssa = B'11: All the 32-bits of address (full address)   |
| WDWM    | B'1000 | (sa)(sd)   | Indicates that stored information of M-bus data tracing in accessing the window area. In the data part following this command, a stored address and stored data are output in the order on pins AUDATA[3:0].                    |
|         |        |            | sa: Indicates the size of a stored address.   |
|         |        |            | B'00: The lower 4 bits of the address   |
|         |        |            | B'01: The lower 8 bits of the address   |
|         |        |            | B'10: The lower 16 bits of the address  |
|         |        |            | B'11: All the 32 bits of address (full address)   |
|         |        |            | sd: Indicates the size of stored data.  |
|         |        |            | B'01: Byte data (8 bits)  |
|         |        |            | B'10: Word data (16 bits)   |
|         |        |            | B'11: Longword data (32 bits)   |
| WDWI    | B'1001 | (sa)(sd)   | Indicates that stored information of I-bus data tracing in accessing the window area. In the data part following this command, a stored address and stored data are output in the order on pins AUDATA[3:0].                    |
|         |        |            | sa: Indicates the size of a stored address.   |
|         |        |            | B'00: The lower 4 bits of the address   |
|         |        |            | B'01: The lower 8 bits of the address   |
|         |        |            | B'10: The lower 16 bits of the address  |
|         |        |            | B'11: All the 32 bits of address (full address)   |
|         |        |            | sd: Indicates the size of stored data.  |
|         |        |            | B'01: Byte data (8 bits)  |
|         |        |            | B'10: Word data (16 bits)   |
|         |        |            | B'11: Longword data (32 bits)   |

| Command | CMD1   | CMD2     | Descriptions  |
|---------|--------|----------|---|
| WDRM    | B'1101 | (sa)(sd) | Indicates that information of M-bus data tracing in reading the window area. In the data part following this command, a read address and read data are output in the order on pins AUDATA[3:0]. |
|         |        |          | sa: Indicates the size of a read address.   |
|         |        |          | B'00: The lower 4 bits of the address   |
|         |        |          | B'01: The lower 8 bits of the address   |
|         |        |          | B'10: The lower 16 bits of the address  |
|         |        |          | B'11: All the 32 bits of address (full address)   |
|         |        |          | sd: Indicates the size of read data.  |
|         |        |          | B'01: Byte data (8 bits)  |
|         |        |          | B'10: Word data (16 bits)   |
|         |        |          | B'11: Longword data (32 bits)   |
| WDRI    | B'1110 | (sa)(sd) | Indicates that information of I-bus data tracing in reading the window area. In the data part following this command, a read address and read data are output in the order on pins AUDATA[3:0]. |
|         |        |          | sa: Indicates the size of a read address.   |
|         |        |          | B'00: The lower 4 bits of the address   |
|         |        |          | B'01: The lower 8 bits of the address   |
|         |        |          | B'10: The lower 16 bits of the address  |
|         |        |          | B'11: All the 32 bits of address (full address)   |
|         |        |          | sd: Indicates the size of read data.  |
|         |        |          | B'01: Byte data (8 bits)  |
|         |        |          | B'10: Word data (16 bits)   |
|         |        |          | B'11: Longword data (32 bits)   |

#### (2) Branch Tracing

The branch tracing function traces events in which the PC changes due to the execution of a branch instruction or the occurrence of an interrupt, and outputs the addresses of a branch source and a branch destination to an external device. Set the BR[1:0] and EN bits in AUCSR and the TREX, TRSB, and TRGN bits in AUECSR to execute the branch tracing of a user program.

Figures 21.2 to 21.4 show examples of branch tracing output.

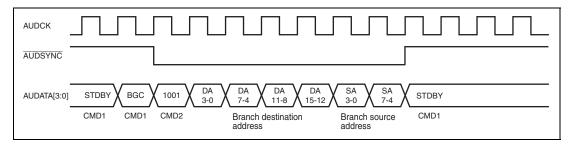


Figure 21.2 Branch Tracing using BGC Command (Branch Destination and Source) (Lower 16 Bits of Branch Destination Address; Lower 8 Bits of Branch Source Address)

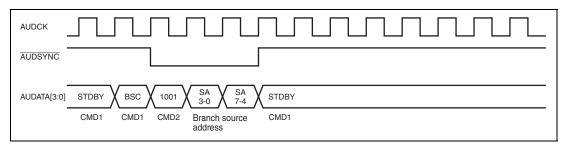


Figure 21.3 Branch Tracing using BSC Command (Branch Source Only)
(Lower 8 Bits of Branch Source Address)

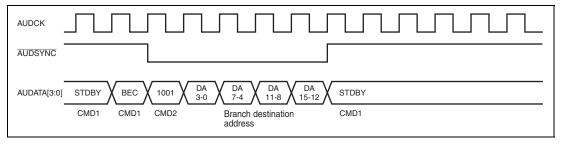


Figure 21.4 Branch Tracing using BEC Command (Branch Destination Only)
(Lower 16 Bits of Branch Destination Address)

When both a branch destination and branch source are output, address differences are compared with the previous branch destination address. When only a branch source is output, address differences are compared with the previous branch source address. When only a branch destination is output, address differences are compared with the previous branch destination address.

In a user program, no branch information is output as long as a branching event (branch instruction execution or an interrupt) does not occur (the STDBY command is output instead). When a branch event occurs while the source and destination addresses are both set to be output, following the command, a branch destination address and a branch source address are output in the order on pins AUDATA[3:0]. The sizes of the addresses (4, 8, 16, or 32 bits) to be output are determined by comparing the addresses with the previously output branch destination address (PFBA).

This algorithm is shown in figure 21.5.

```
Definition:
PFBA: Previously accessed address of branch destination (all 32 bits stored)
      (when only branch source is traced, the branch source address is stored)
CDA: Branch destination address to be output
CSA: Branch source address to be output
if(PFBA[31:0] == 0x00000000) {output32bit(CDA[31:0]);}
else if (PFBA[31:4] == CDA[31:4]) {output4bit(CDA[3:0]);}
else if (PFBA[31:8] == CDA[31:8]) {output8bit(CDA[7:0]);}
else if(PFBA[31:16] == CDA[31:16]) {output16bit(CDA[15:0]);}
else {output32bit(CDA[31:0]);}
if(PFBA[31:0] == 0x00000000) {output32bit(CSA[31:0]);}
else if (PFBA[31:4] == CSA[31:4]) {output4bit(CSA[3:0]);}
else if(PFBA[31:8] == CSA[31:8]) {output8bit(CSA[7:0]);}
else if(PFBA[31:16] == CSA[31:16]) {output16bit(CSA[15:0]);}
else {output32bit(CSA[31:0]);}
PFBA = CDA; /* update PFBA */
```

Figure 21.5 Branch Tracing Algorithm

The use of this algorithm can substantially reduce the amount of traced data to be output, window data tracing information is accumulated as long as there is available space in the FIFO. The operation when the FIFO is full depends on the TM bit (tracing mode) in AUCSR.

#### Branch Source Address

The address to which a branch source address (TSA) points depends on the type of branches as follows:

— Branch instructions (general and subroutine branches) The TSA points to the address of the branch instruction.

- Instruction-asynchronous exception (16-bit and 32-bit instructions)
   The TSA points to the address of the instruction that has been replaced by the exception handling.
- Instruction-synchronous exception (the TRAPA and RTE instructions)
   The TSA points to the address of the instruction.

Table 21.7 Definition of Branch Source Addresses

| Туре | General<br>Branching | Subroutine<br>Branching | Instruction<br>Asynchronous<br>Exception<br>(16 Bits) | Instruction<br>Asynchronous<br>Exception<br>(32 Bits) | Instruction<br>Synchronous<br>Exception |
|------|----------------------|-------------------------|---|---|---|
| 1000 | BRA                  | BSR                     | NOP*1   | 32-bit instruction*1                                  | TRAPA                                   |
| 1002 | NOP                  | NOP                     | NOP*2   |   | NOP                                     |
| 1004 | NOP                  | NOP                     | NOP   | NOP*2   | NOP                                     |
| 1006 | NOP                  | NOP                     | NOP   | NOP   | NOP                                     |
| TSA  | 1000                 | 1000                    | 1002  | 1004  | 1000                                    |

Notes: 1. Indicates that the instruction has been executed.

2. Indicates that the instruction has been replaced by the exception handling.

#### **(3)** Window Data Tracing

The window data tracing function outputs traced information in accessing to a memory area (called window) specified by two address pointers. AUD-II can handle two window areas: window A specified by AUWASR and AUWAER and window B specified by AUWBSR and AUWBER. By setting the EN bit to 1 after setting the WA[1:0] and WB[1:0] bits in AUCSR, memory access information in a user program is traced.

Figure 21.6 shows an example of window data tracing.

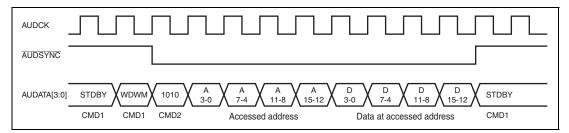


Figure 21.6 Window Data Tracing using WDWM Command (M Bus, Lower 16 Bits of Address, Word-Size Data)

When data in a window area is accessed in a user program, the command, accessed address, and the data at the accessed address are output in the order on pins AUDATA[3:0]. In a way similar to branch tracing, the address sizes (4, 8, 16, or 32 bits) to be output are determined by comparing the two addresses with the previously output address (PFDA). The data at the accessed address is output according to the access size (8, 16, or 32).

This algorithm is shown in figure 21.7.

```
Definition:
PFDA: Previously accessed address (all 32 bits stored)
CDA: Data to be output
/* PFDA : initial value = H'00000000 */
if(PFDA[31:0] == 0x000000000) \{output32bit(CDA[31:0]);\}
else if(PFDA[31:4] == CDA[31:4]) {output4bit(CDA[3:0]);}
else if(PFDA[31:8] == CDA[31:8]) {output8bit(CDA[7:0]);}
else if(PFDA[31:16] == CDA[31:16]) {output16bit(CDA[15:0]);}
else {output32bit(CDA[31:0]);}
PFDA = CDA; /* update PFDA */
```

Figure 21.7 Window Data Tracing Algorithm

The use of this algorithm can substantially reduce the amount of traced data to be output. Branch tracing information is accumulated as long as there is available space in the FIFO. The operation when the FIFO is full depends on the TM bit (tracing mode) in AUCSR.

#### (4) Realtime Tracing Mode

Realtime tracing mode outputs various data traced by branch tracing or window data tracing in realtime to an external device via pins AUDATA[3:0]. Setting the TM bit in AUCSR to 1 selects this mode.

In this mode, the CPU operates in the same way as when the tracing function is not used, even during output of trace data. The traced events are stored as long as there is available space in the FIFO of AUD-II, and are output to an external device. When the FIFO is full, the events that should be traced are ignored, and they are not output externally. However, any missed event is notified to an external device by the LOST command. In the cycle following the LOST command output cycle, the AUDSYNC signal is driven low.

If multiple events occur simultaneously, corresponding information is stored in the FIFO in the following order: window data tracing (M bus > I bus) and branch tracing.

Figure 21.8 shows an example where multiple traced information is lost between two branch trace events that are output. Figure 21.9 shows an example of LOST and STDBY states following the LOST command.

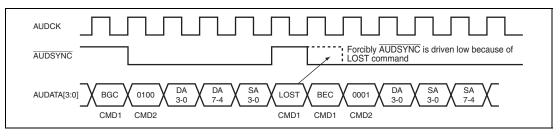


Figure 21.8 Example Where Trace Data is Lost during Realtime Tracing

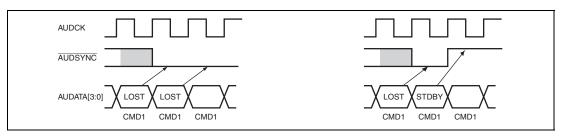


Figure 21.9 Example of LOST and STDBY States Following LOST Command

### (5) Full Tracing Mode

In full tracing mode, all traced data is output externally, without losing it. This mode is selected by setting the TM bit in AUCSR to 0. Various types of traced information in full tracing mode are output externally through the FIFO in AUD-II. If the FIFO becomes full, the CPU stops operation until the traced data in the FIFO is output to an external device.

In this a case, unlike the realtime tracing mode, traced data is not lost. However, if the CPU temporarily stops operation because the FIFO is full, the LOST command indicating the temporary halting of the CPU is output. In a way similar to the realtime tracing mode, in the cycle following the LOST command output cycle, the AUDSYNC signal is driven low as shown in figure 21.9.

#### (6) Address Comparison and Output of Address Differences

For the efficient use of the narrow-bit-width AUDATA output, only the necessary lower bits of traced information are output in the address part. The lower bits are obtained by comparing the traced address with the previously output address stored in PFBA and PFDA.

The sizes (4, 8, 16, or 32 bits) of the branch destination and source addresses are determined by comparing the two addresses with PFBA. Similarly, the address to be output by window data tracing is determined by comparing the two addresses with PFDA. PFBA and PFDA used to compare addresses are updated each time the compared traced information is output.

PFBA and PFDA are initialized to H'00000000 (called the disabled state) when the EN bit in AUCSR is changed from 0 to 1, or when the output counter overflows. When PFBA and PFDA are disabled, 32 bits of the address are output in the address part of traced information. The output counter is incremented each time a trace event occurs, and disables PFBA and PFDA once per 128 times a trace event occurs. In addition, the output counter is cleared to 0 when the EN bit in AUCSR is changed from 0 to 1. The output counter functions can be used for recovery of the corrupted traced information.

Figure 21.10 shows an example of address comparison.

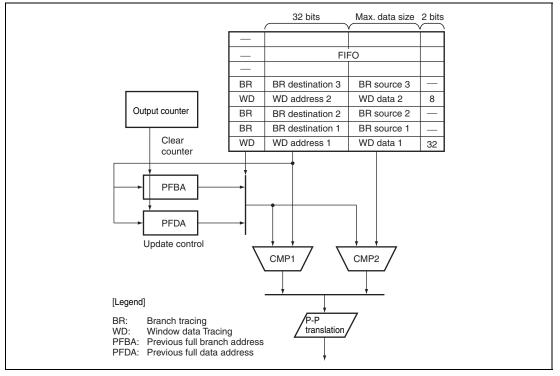


Figure 21.10 Example of Address Comparison

- 1. First, WD address 1 and PFDA are matched and compared in CMP1. Based on the results of the comparison, the lower part (4, 8, 16, or 32 bits) of WD address 1 is output. Next, WD data 1, 32-bit long, is output in its native length. Subsequently, PFDA is updated to WD address 1.
- 2. In the next step, BR destination 1 and PFBA are compared in CMP1, and BR source 1 and PFBA are compared in CMP2. Based on the results of the comparison in CMP1, the lower part (4, 8, 16, or 32 bits) of BR destination 1 is output. Next, the lower part (4, 8, 16, or 32 bits) of BR source 1 is output based on the results of comparison in CMP2. Subsequently, PFBA is updated to BR destination 1.
- 3. BR destination 2 and PFBA are compared in CMP1, and BR source 2 and PFBA are compared in CMP2. PFBA holds BR destination 1, which was stored in step 2 above. According to the results of the comparison in CMP1, the lower part (4, 8, 16, or 32 bits) of BR destination 2 is output. Next, the lower part (4, 8, 16, or 32 bits) of BR source 2 is output according to the results of comparison in CMP2. Subsequently, PFBA is updated to BR destination 2.

- 4. WD address 2 and PFDA are compared in CMP1. PFDA holds WD address 1, which was stored in step 1 above. Based on the results of the comparison, the lower part (4, 8, 16, or 32 bits) of WD address 2 is output. Next, WD data 2, which is 8 bits long, is output in its native bit length. Subsequently, PFDA is updated to WD address 2.
- 5. BR destination 3 and PFBA are compared in CMP1, and BR source 3 and PFBA are compared in CMP2. PFBA holds BR destination 2, which was stored in step 3 above. Based on the results of the comparison in CMP1, the lower part (4, 8, 16, or 32 bits) of BR destination 3 is output. Next, based on the results of the comparison in CMP2, the lower part (4, 8, 16, or 32 bits) of BR source 3 is output. Subsequently, PFBA is updated to BR destination 3.
- 6. If the output counter overflows, the values in PFBA and PFDA are nullified. The branch destination, the branch source, and the data address part in the trace information are always output in a 32-bit length.

#### 21.3.9 **Usage Notes on AUD Tracing Mode**

#### Rules on Initialization of AUD Tracing Mode **(1)**

AUD-II is initialized according to the following conditions.

The AUCSR and AUECSR registers are initialized to H'0000 by a power-on reset, a low level input on the AUDRST pin, and a module standby. At the same time, AUDATA[3:0], AUDSYNC, and AUDCK output H'0, B'1, and B'1, respectively.

- Power-on reset
- A low level input on the  $\overline{AUDRST}$  pin
- Module standby

Note that the module standby mode is not entered under the following conditions.

- When the EN bit in AUCSR is 1
- When the traced data is in the FIFO
- The EN bit in AUCSR changes from 0 to 1.

The FIFO of AUD-II is cleared (emptied), and PFBA, PFDA, and the output counter are initialized.

The output counter overflows.

The values in PFBA and PFDA are initialized (H'00000000).

#### (2) Rules on AUDCK

- When the EN bit in AUCSR is 1, the clock signal is output on the AUDCK pin. The clock signal is not output when the EN bit is 0. However, if the FIFO contains valid data, this state is attained after all data has been transmitted.
- The CK[1:0] bits in AUCSR must be changed when the EN bit in AUCSR is 0.
- The frequency of the AUDCK clock must not exceed the input frequency from the EXTAL pin. Settable AUDCK clock ratios are listed below:

Table 21.8 Settable AUDCK Clock Ratios

| PLL Multiplication Ratio of Internal Clock (φ) | Allowable AUDCK Clock Ratio |
|--|-----------------------------|
| ×4   | 1/4, 1/8, 1/10              |
| ×6, ×8   | 1/8, 1/10                   |
| ×10  | 1/10                        |

#### (3) Writing to AUD-II Register

- The AUD-II registers are written to through the I bus. Therefore, in a cycle immediately after an AUD-II register is rewritten to change tracing conditions, tracing may not be performed according to the modified conditions. To ensure that the conditions in the AUD-II register are changed, execute more than five NOP instructions after reading the rewritten register once. After that tracing conditions will be reflected in the register.
- The AUD-II register must be changed when the EN bit in AUCSR is 0.

#### (4) Other Notes

- If the FIFO of AUD-II contains valid data and the tracing function is disabled, the FIFO
  assumes the disabled state after the remaining data is output. The tracing function should not
  be enabled when the data remaining in the FIFO is being output. The tracing function of AUDII should be enabled only after confirming that no data is being output on AUD-II.
- The AUD tracing function can be used in sleep mode.

#### 21.4 **RAM Monitoring Mode**

In RAM monitoring mode, all memory-mapped modules connected to internal or external buses can be read from or written to. The contents of the on-chip RAM can be monitored or tuned.

#### 21.4.1 Communication Protocol

AUD-II reads data on the AUDATA pins at an assertion of the AUDSYNC signal. Input the AUDATA signals in the following format.

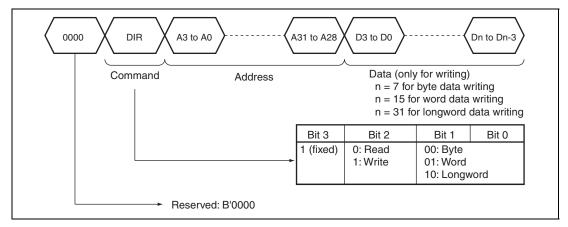


Figure 21.11 AUDATA Input Format

#### 21.4.2 **Operations**

To use the RAM monitoring function, negate the AUDMD signal while the AUDRST signal is asserted. Then negate the AUDRST signal. After that, AUD-II enters the RAM monitoring mode. Figure 21.12 shows an example of read operation and figure 21.13 shows an example of write operation.

Assert the AUDSYNC signal before data is input. Input a command, address, and data (required only for writing) in the format shown in figure 21.11. AUD-II starts reading from or writing to the specified address. AUD-II outputs the not-ready code (B'0000) during the internal processing and the ready code (B'0001) on completion of the internal processing (see figures 21.12 and 21.13). Table 21.9 shows the meaning of the code.

Apr 01, 2014

For reading, negate the AUDSYNC signal when detecting the ready code. AUD-II outputs the specified size of data (see figure 21.12). When an undefined command (i.e., other than a DIR command) is input, AUD-II handles the command as an error. In this case, AUD-II does not read or write and sets bit 1 in the code shown in table 21.9 to 1.

If a command specified by the DIR command causes a bus error, AUD-II does not execute read or write specified by the command and sets bit 2 in the code shown in table 21.9 to 1 (see figure 21.14).

The bus errors are shown below.

- When an address of 4n + 1 or 4n + 3 is accessed in words
- When an address of 4n + 1, 4n + 2, or 4n + 3 is accessed in longwords
- When an address in the external space is accessed in single-chip mode

#### **Table 21.9** Meaning of Each Bit of Code

| Bit | Description                     |
|-----|---------------------------------|
| 3   | 0 (fixed)                       |
| 2   | 0: Normal state                 |
|     | 1: A bus error has occurred     |
| 1   | 0: Normal state                 |
|     | 1: A command error has occurred |
| 0   | 0: Not ready                    |
|     | 1: Ready                        |

Page 1047 of 1824

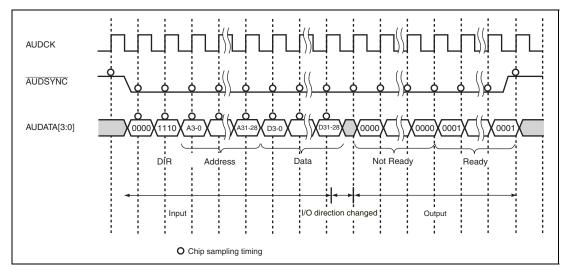


Figure 21.12 Example Operation of Byte Read

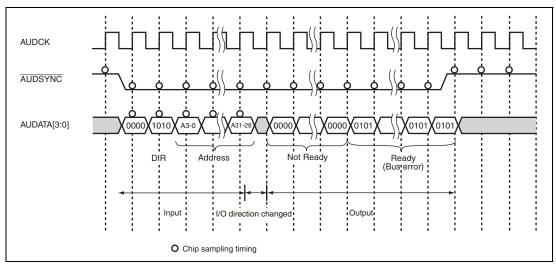


Figure 21.13 Example Operation of Longword Write

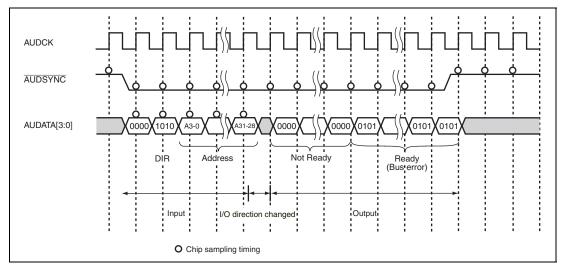


Figure 21.14 Example Operation of Error (Longword Read)

#### 21.4.3 Notes on RAM Monitoring Mode

#### (1) Rules on Initialization in RAM Monitoring

Buffers and internal states of AUD-II are initialized under the following conditions.

- When a power-on reset is input
- When a low level is input on the  $\overline{AUDRST}$  pin
- When a transition to the module standby mode is made

#### (2) Rules on AUDCK

AUDCK is for an external clock input in this mode. The frequency of the external clock input on the AUDCK pin must be less than that of the input clock from the EXTAL pin.

#### (3) Other Notes

Hold the AUDSYNC signal low until the ready code is returned after inputting a command on the AUDATA pins.

The RAM monitoring function can be used in sleep mode.

# Section 22 Pin Function Controller (PFC)

The pin function controller (PFC) consists of the registers that control the functions and direction (I/O) of the multiplex pins. The multiplex pins of this LSI are summarized in tables 22.1 to 22.11.

**Table 22.1** Multiplex Pin List (Port A)

| Function 1<br>(Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4<br>(Related Module) |
|--------------------------------|-----------------------------|-----------------------------|--------------------------------|
| PA0 (port)                     | A0 (BSC)                    | _                           | _                              |
| PA1 (port)                     | A1 (BSC)                    | _                           | _                              |
| PA2 (port)                     | A2 (BSC)                    | _                           | _                              |
| PA3 (port)                     | A3 (BSC)                    | _                           | _                              |
| PA4 (port)                     | A4 (BSC)                    | _                           | _                              |
| PA5 (port)                     | A5 (BSC)                    | _                           | _                              |
| PA6 (port)                     | A6 (BSC)                    | _                           | _                              |
| PA7 (port)                     | A7 (BSC)                    | _                           | _                              |
| PA8 (port)                     | A8 (BSC)                    | _                           | _                              |
| PA9 (port)                     | A9 (BSC)                    | _                           | _                              |
| PA10 (port)                    | A10 (BSC)                   | _                           | _                              |
| PA11 (port)                    | A11 (BSC)                   | _                           | _                              |
| PA12 (port)                    | A12 (BSC)                   | _                           | _                              |
| PA13 (port)                    | A13 (BSC)                   | _                           | _                              |
| PA14 (port)                    | A14 (BSC)                   | _                           | _                              |
| PA15 (port)                    | A15 (BSC)                   | _                           | _                              |

Table 22.2 Multiplex Pin List (Port B)

| Function 1<br>(Related Module) | Function 2<br>(Related Module) | Function 3<br>(Related Module) | Function 4<br>(Related Module) |
|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| PB0 (port)                     | A16 (BSC)                      | MOSIA (RSPI)                   | _                              |
| PB1 (port)                     | A17 (BSC)                      | MISOA (RSPI)                   | _                              |
| PB2 (port)                     | A18 (BSC)                      | MOSIB (RSPI)                   | _                              |
| PB3 (port)                     | A19 (BSC)                      | MISOB (RSPI)                   | _                              |
| PB4 (port)                     | A20 (BSC)                      | CTx_B (RCAN-TL1)               | TIF6 (ATU-III)                 |
| PB5 (port)                     | A21 (BSC)                      | CRx_B (RCAN-TL1)               | TIF7 (ATU-III)                 |
| PB6 (port)                     | WE0 (BSC)                      | _                              | _                              |
| PB7 (port)                     | WE1 (BSC)                      | _                              | _                              |
| PB8 (port)                     | WAIT (BSC)                     | TOE20 (ATU-III)                | _                              |
| PB9 (port)                     | RD (BSC)                       | _                              | _                              |
| PB10 (port)                    | CS0 (BSC)                      | _                              | _                              |
| PB11 (port)                    | CS1 (BSC)                      | TOE21 (ATU-III)                | _                              |
| PB12 (port)                    | CS2 (BSC)                      | RSPCKA (RSPI)                  | _                              |
| PB13 (port)                    | CS3 (BSC)                      | RSPCKB (RSPI)                  | _                              |
| PB14 (port)                    | RD/WR (BSC)                    | _                              | _                              |

**Table 22.3** Multiplex Pin List (Port C)

| Function 1<br>(Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4<br>(Related Module) |
|--------------------------------|-----------------------------|-----------------------------|--------------------------------|
| PC0 (port)                     | D0 (BSC)                    | _                           | _                              |
| PC1 (port)                     | D1 (BSC)                    | _                           | _                              |
| PC2 (port)                     | D2 (BSC)                    | _                           | _                              |
| PC3 (port)                     | D3 (BSC)                    | _                           | _                              |
| PC4 (port)                     | D4 (BSC)                    | _                           | _                              |
| PC5 (port)                     | D5 (BSC)                    | _                           | _                              |
| PC6 (port)                     | D6 (BSC)                    | _                           | _                              |
| PC7 (port)                     | D7 (BSC)                    | _                           | _                              |
| PC8 (port)                     | D8 (BSC)                    | _                           | _                              |
| PC9 (port)                     | D9 (BSC)                    | _                           | _                              |
| PC10 (port)                    | D10 (BSC)                   | _                           | _                              |
| PC11 (port)                    | D11 (BSC)                   | _                           | _                              |
| PC12 (port)                    | D12 (BSC)                   | _                           | _                              |
| PC13 (port)                    | D13 (BSC)                   | _                           | _                              |
| PC14 (port)                    | D14 (BSC)                   | _                           | _                              |
| PC15 (port)                    | D15 (BSC)                   | _                           | _                              |

Table 22.4 Multiplex Pin List (Port D)

| Function 1<br>(Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4<br>(Related Module) |
|--------------------------------|-----------------------------|-----------------------------|--------------------------------|
| PD0 (port)                     | TIOC00 (ATU-III)            | TIOC31 (ATU-III)            | _                              |
| PD1 (port)                     | TIOC01 (ATU-III)            | TOE20 (ATU-III)             | _                              |
| PD2 (port)                     | TIOC02 (ATU-III)            | TOE21 (ATU-III)             | TOE52 (ATU-III)                |
| PD3 (port)                     | TIOC03 (ATU-III)            | TOE22 (ATU-III)             | TOE53 (ATU-III)                |
| PD4 (port)                     | TIOC10 (ATU-III)            | TIOC32 (ATU-III)            | TOE52 (ATU-III)                |
| PD5 (port)                     | TIOC11 (ATU-III)            | TOE23 (ATU-III)             | TOE40 (ATU-III)                |
| PD6 (port)                     | TIOC12 (ATU-III)            | _                           | TOE41 (ATU-III)                |
| PD7 (port)                     | TIOC13 (ATU-III)            | _                           | TOE42 (ATU-III)                |
| PD8 (port)                     | TIOC20 (ATU-III)            | TIOC33 (ATU-III)            | TOE53 (ATU-III)                |
| PD9 (port)                     | TIOC21 (ATU-III)            | TIF0B (ATU-III)             | TOE43 (ATU-III)                |
| PD10 (port)                    | TIOC22 (ATU-III)            | TIF1B (ATU-III)             | TOE50 (ATU-III)                |
| PD11 (port)                    | TIOC23 (ATU-III)            | TIF2B (ATU-III)             | TOE51 (ATU-III)                |
| PD12 (port)                    | TCLKA (ATU-III)             | TIOC41 (ATU-III)            | TIJ0 (ATU-III)                 |
| PD13 (port)                    | TCLKB (ATU-III)             | _                           | TIJ1 (ATU-III)                 |

**Table 22.5** Multiplex Pin List (Port E)

| Function 1<br>(Related Module) | Function 2<br>(Related Module) | Function 3 (Related Module) | Function 4<br>(Related Module) |
|--------------------------------|--------------------------------|-----------------------------|--------------------------------|
| PE0 (port)                     | TIA00 (ATU-III)                | _                           | _                              |
| PE1 (port)                     | TIA01 (ATU-III)                | TIOC42 (ATU-III)            | TIOC40 (ATU-III)               |
| PE2 (port)                     | TIA02 (ATU-III)                | TIOC43 (ATU-III)            | TIOC30 (ATU-III)               |
| PE3 (port)                     | TIA03 (ATU-III)                | _                           | _                              |
| PE4 (port)                     | TIA04 (ATU-III)                | _                           | _                              |
| PE5 (port)                     | TIA05 (ATU-III)                | _                           | _                              |
| PE6 (port)                     | TOE00 (ATU-III)                | CTx_B (RCAN-TL1)            | _                              |
| PE7 (port)                     | TOE01 (ATU-III)                | CRx_B (RCAN-TL1)            | _                              |
| PE8 (port)                     | TOE02 (ATU-III)                | _                           | _                              |
| PE9 (port)                     | TOE03 (ATU-III)                | _                           | _                              |
| PE10 (port)                    | TOE10 (ATU-III)                | _                           | _                              |
| PE11 (port)                    | TOE11 (ATU-III)                | _                           | _                              |
| PE12 (port)                    | TOE12 (ATU-III)                | _                           | _                              |
| PE13 (port)                    | TOE13 (ATU-III)                | _                           | _                              |

**Table 22.6** Multiplex Pin List (Port F)

| Function 1<br>(Related Module) | Function 2<br>(Related Module) | Function 3 (Related Module) | Function 4<br>(Related Module) |
|--------------------------------|--------------------------------|-----------------------------|--------------------------------|
| PF0 (port)                     | TOD00B (ATU-III)               | _                           | TIF6 (ATU-III)                 |
| PF1 (port)                     | TOD01B (ATU-III)               | _                           | TIF7 (ATU-III)                 |
| PF2 (port)                     | TOD02B (ATU-III)               | _                           | TIF8 (ATU-III)                 |
| PF3 (port)                     | TOD03B (ATU-III)               | _                           | TIF9 (ATU-III)                 |
| PF4 (port)                     | TOD10B (ATU-III)               | _                           | TIF10 (ATU-III)                |
| PF5 (port)                     | TOD11B (ATU-III)               | _                           | TIF11 (ATU-III)                |
| PF6 (port)                     | TOD12B (ATU-III)               | _                           | TIF12 (ATU-III)                |
| PF7 (port)                     | TOD13B (ATU-III)               | _                           | TIF13 (ATU-III)                |
| PF8 (port)                     | TOD20B (ATU-III)               | _                           | TIF14 (ATU-III)                |
| PF9 (port)                     | TOD21B (ATU-III)               | _                           | TIF15 (ATU-III)                |
| PF10 (port)                    | TOD22B (ATU-III)               | _                           | TIF16 (ATU-III)                |
| PF11 (port)                    | TOD23B (ATU-III)               | _                           | TIF17 (ATU-III)                |
| PF12 (port)                    | TOD30B (ATU-III)               | _                           | TIF18 (ATU-III)                |
| PF13 (port)                    | TOD31B (ATU-III)               | _                           | TIF19 (ATU-III)                |
| PF14 (port)                    | TOD32B (ATU-III)               | CTx_B (RCAN-TL1)            | TxD_A (SCI)                    |
| PF15 (port)                    | TOD33B (ATU-III)               | CRx_B (RCAN-TL1)            | RxD_A (SCI)                    |

**Table 22.7** Multiplex Pin List (Port G)

| Function 1<br>(Related Module) | Function 2<br>(Related Module) | Function 3<br>(Related Module) | Function 4<br>(Related Module) |
|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| PG0 (port)                     | TOD00A (ATU-III)               | SSLA0 (RSPI)                   | _                              |
| PG1 (port)                     | TOD01A (ATU-III)               | SSLA1 (RSPI)                   | _                              |
| PG2 (port)                     | TOD02A (ATU-III)               | SSLA2 (RSPI)                   | _                              |
| PG3 (port)                     | TOD03A (ATU-III)               | SSLA3 (RSPI)                   | _                              |
| PG4 (port)                     | TOD10A (ATU-III)               | SSLA4 (RSPI)                   | SSLB3 (RSPI)                   |
| PG5 (port)                     | TOD11A (ATU-III)               | SSLA5 (RSPI)                   | SSLC3 (RSPI)                   |
| PG6 (port)                     | TOD12A (ATU-III)               | SSLB0 (RSPI)                   | _                              |
| PG7 (port)                     | TOD13A (ATU-III)               | SSLB1 (RSPI)                   | _                              |
| PG8 (port)                     | TOD20A (ATU-III)               | SSLB2 (RSPI)                   | TIF6 (ATU-III)                 |
| PG9 (port)                     | TOD21A (ATU-III)               | SSLC0 (RSPI)                   | TIF7 (ATU-III)                 |
| PG10 (port)                    | TOD22A (ATU-III)               | SSLC1 (RSPI)                   | TIF8 (ATU-III)                 |
| PG11 (port)                    | TOD23A (ATU-III)               | SSLC2 (RSPI)                   | TIF9 (ATU-III)                 |
| PG12 (port)                    | TOD30A (ATU-III)               | SSLA4 (RSPI)                   | TIF10 (ATU-III)                |
| PG13 (port)                    | TOD31A (ATU-III)               | SSLA5 (RSPI)                   | TIF11 (ATU-III)                |
| PG14 (port)                    | TOD32A (ATU-III)               | SSLA6 (RSPI)                   | TIF12 (ATU-III)                |
| PG15 (port)                    | TOD33A (ATU-III)               | SSLA7 (RSPI)                   | TIF13 (ATU-III)                |

Table 22.8 Multiplex Pin List (Port H)

| Function 1<br>(Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4<br>(Related Module) |
|--------------------------------|-----------------------------|-----------------------------|--------------------------------|
| PH0 (port)                     | _                           | ADTRG_A (ADC)               | TIF0A (ATU-III)                |
| PH1 (port)                     | _                           | ADTRG_B (ADC)               | TIF1A (ATU-III)                |
| PH2 (port)                     | _                           | _                           | TIF2A (ATU-III)                |
| PH3 (port)                     | _                           | _                           | TIF3 (ATU-III)                 |
| PH4 (port)                     | _                           | _                           | TIF4 (ATU-III)                 |
| PH5 (port)                     | _                           | _                           | TIF5 (ATU-III)                 |

Table 22.9 Multiplex Pin List (Port J)

| Function 1<br>(Related Module) | Function 2<br>(Related Module) | Function 3 (Related Module) | Function 4<br>(Related Module)   |
|--------------------------------|--------------------------------|-----------------------------|----------------------------------|
| PJ0 (port)                     | TxD_A (SCI)                    | CTx_A (RCAN-TL1)            | CTx_A & CTx_B<br>(RCAN-TL1)      |
| PJ1 (port)                     | RxD_A (SCI)                    | CRx_A (RCAN-TL1)            | CRx_A & CRx_B<br>(RCAN-TL1)      |
| PJ2 (port)                     | TxD_A (SCI)                    | CTx_C (RCAN-TL1)            | CTx_A & CTx_B & CTx_C (RCAN-TL1) |
| PJ3 (port)                     | RxD_A (SCI)                    | CRx_C (RCAN-TL1)            | CRx_A & CRx_B & CRx_C (RCAN-TL1) |
| PJ4 (port)                     | SCK_A (SCI)                    | ADEND_B (ADC)               | TIJ0 (ATU-III)                   |
| PJ5 (port)                     | TxD_A (SCI)                    | _                           | _                                |
| PJ6 (port)                     | RxD_A (SCI)                    | _                           | _                                |
| PJ7 (port)                     | SCK_B (SCI)                    | ADEND_A (ADC)               | TIJ1 (ATU-III)                   |
| PJ8 (port)                     | TxD_B (SCI)                    | _                           | _                                |
| PJ9 (port)                     | RxD_B (SCI)                    | _                           | _                                |

Table 22.10 Multiplex Pin List (Port K)

| Function 1<br>(Related Module) | Function 2 (Related Module) | Function 3 Function 4 (Related Module) (Related Module) |   |
|--------------------------------|-----------------------------|---|---|
| PK0 (port)                     | SCK_C (SCI)                 | RSPCKA (RSPI) UBCTRG (UBC)                              |   |
| PK1 (port)                     | TxD_C (SCI)                 | MOSIA (RSPI)  | _ |
| PK2 (port)                     | RxD_C (SCI)                 | MISOA (RSPI)  | _ |
| PK3 (port)                     | SCK_D (SCI)                 | RSPCKB (RSPI)   | _ |
| PK4 (port)                     | TxD_D (SCI)                 | MOSIB (RSPI)  | _ |
| PK5 (port)                     | RxD_D (SCI)                 | MISOB (RSPI)  | _ |
| PK6 (port)                     | SCK_E (SCI)                 | _   | _ |
| PK7 (port)                     | TxD_E (SCI)                 | _   | _ |
| PK8 (port)                     | RxD_E (SCI)                 | _   | _ |
| PK9 (port)                     | _                           | RSPCKC (RSPI)   | _ |
| PK10 (port)                    | _                           | MOSIC (RSPI)  | _ |
| PK11 (port)                    | _                           | MISOC (RSPI)  | _ |

Table 22.11 Multiplex Pin List (Port L)

| Function 1<br>(Related Module) | Function 2<br>(Related Module) | Function 3 (Related Module) | Function 4<br>(Related Module) |
|--------------------------------|--------------------------------|-----------------------------|--------------------------------|
| PL0 (port)                     | _                              | ĪRQ0 (INTC)                 | _                              |
| PL1 (port)                     | TOE20 (ATU-III)                | IRQ1 (INTC)                 | POD (port)                     |
| PL2 (port)                     | TOE21 (ATU-III)                | ĪRQ2 (INTC)                 | _                              |
| PL3 (port)                     | TOE22 (ATU-III)                | ĪRQ3 (INTC)                 | _                              |
| PL4 (port)                     | TOE23 (ATU-III)                | IRQ4 (INTC)                 | _                              |
| PL5 (port)                     | TOE30 (ATU-III)                | ĪRQ5 (INTC)                 | _                              |
| PL6 (port)                     | TOE31 (ATU-III)                | ĪRQ6 (INTC)                 | _                              |
| PL7 (port)                     | TOE32 (ATU-III)                | ĪRQ7 (INTC)                 | _                              |
| PL8 (port)                     | TOE33 (ATU-III)                | _                           | _                              |

# 22.1 Register Descriptions

The PFC registers are summarized in table 22.12.

**Table 22.12 Register Configuration** 

| Register Name             | Abbreviation | R/W | Initial Value  | Address    | Access Size |
|---------------------------|--------------|-----|--|------------|-------------|
| Port A I/O register       | PAIOR        | R/W | H'0000   | H'FFFE3806 | 8, 16       |
| Port A control register 4 | PACR4        | R/W | H'1111* <sup>1</sup><br>H'1111* <sup>2</sup><br>H'1111* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3810 | 8, 16, 32   |
| Port A control register 3 | PACR3        | R/W | H'1111* <sup>1</sup><br>H'1111* <sup>2</sup><br>H'1111* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3812 | 8, 16       |
| Port A control register 2 | PACR2        | R/W | H'1111* <sup>1</sup><br>H'1111* <sup>2</sup><br>H'1111* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3814 | 8, 16, 32   |
| Port A control register 1 | PACR1        | R/W | H'1111* <sup>1</sup><br>H'1111* <sup>2</sup><br>H'1111* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3816 | 8, 16       |
| Port B I/O register       | PBIOR        | R/W | H'0000   | H'FFFE3886 | 8, 16       |
| Port B control register 4 | PBCR4        | R/W | H'0400* <sup>1</sup><br>H'0400* <sup>2</sup><br>H'0400* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3890 | 8, 16, 32   |
| Port B control register 3 | PBCR3        | R/W | H'0110* <sup>1</sup><br>H'0110* <sup>2</sup><br>H'0112* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3892 | 8, 16       |
| Port B control register 2 | PBCR2        | R/W | H'0122* <sup>1</sup><br>H'1122* <sup>2</sup><br>H'1122* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3894 | 8, 16, 32   |
| Port B control register 1 | PBCR1        | R/W | H'2222*1<br>H'2222*2<br>H'2222*3<br>H'0000*4   | H'FFFE3896 | 8, 16       |

| Register Name             | Abbreviation | R/W | Initial Value  | Address    | Access Size |
|---------------------------|--------------|-----|--|------------|-------------|
| Port C I/O register       | PCIOR        | R/W | H'0000   | H'FFFE3906 | 8, 16       |
| Port C control register 4 | PCCR4        | R/W | H'0000* <sup>1</sup><br>H'1111* <sup>2</sup><br>H'1111* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3910 | 8, 16, 32   |
| Port C control register 3 | PCCR3        | R/W | H'0000* <sup>1</sup><br>H'1111* <sup>2</sup><br>H'1111* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3912 | 8, 16       |
| Port C control register 2 | PCCR2        | R/W | H'1111* <sup>1</sup><br>H'1111* <sup>2</sup><br>H'1111* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3914 | 8, 16, 32   |
| Port C control register 1 | PCCR1        | R/W | H'1111* <sup>1</sup><br>H'1111* <sup>2</sup><br>H'1111* <sup>3</sup><br>H'0000* <sup>4</sup> | H'FFFE3916 | 8, 16       |
| Port D I/O register       | PDIOR        | R/W | H'0000   | H'FFFFC808 | 8, 16       |
| Port D control register 2 | PDCR2        | R/W | H'0000   | H'FFFFC80C | 8, 16, 32   |
| Port D control register 1 | PDCR1        | R/W | H'0000   | H'FFFFC80E | 8, 16       |
| Port E I/O register       | PEIOR        | R/W | H'0000   | H'FFFFC818 | 8, 16       |
| Port E control register 2 | PECR2        | R/W | H'0000   | H'FFFFC81C | 8, 16, 32   |
| Port E control register 1 | PECR1        | R/W | H'0000   | H'FFFFC81E | 8, 16       |
| Port F I/O register       | PFIOR        | R/W | H'0000   | H'FFFFC82A | 8, 16       |
| Port F control register 2 | PFCR2        | R/W | H'0000   | H'FFFFC82C | 8, 16, 32   |
| Port F control register 1 | PFCR1        | R/W | H'0000   | H'FFFFC82E | 8, 16       |
| Port G I/O register       | PGIOR        | R/W | H'0000   | H'FFFFC83C | 8, 16       |
| Port G control register 2 | PGCR2        | R/W | H'0000   | H'FFFFC840 | 8, 16, 32   |
| Port G control register 1 | PGCR1        | R/W | H'0000   | H'FFFFC842 | 8, 16       |
| Port H I/O register       | PHIOR        | R/W | H'0000   | H'FFFFC854 | 8, 16       |
| Port H control register   | PHCR         | R/W | H'0000   | H'FFFFC858 | 8, 16       |
| Port J I/O register       | PJIOR        | R/W | H'0000   | H'FFFFC86C | 8, 16       |
| Port J control register 2 | PJCR2        | R/W | H'0000   | H'FFFFC870 | 8, 16, 32   |
| Port J control register 1 | PJCR1        | R/W | H'0000   | H'FFFFC872 | 8, 16       |

| Register Name             | Abbreviation | R/W | Initial Value | Address    | Access Size |
|---------------------------|--------------|-----|---------------|------------|-------------|
| Port K I/O register       | PKIOR        | R/W | H'0000        | H'FFFFC88C | 8, 16       |
| Port K control register 2 | PKCR2        | R/W | H'0000        | H'FFFFC890 | 8, 16, 32   |
| Port K control register 1 | PKCR1        | R/W | H'0000        | H'FFFFC892 | 8, 16       |
| Port L I/O register       | PLIOR        | R/W | H'0000        | H'FFFFC8A8 | 8, 16       |
| Port L control register 2 | PLCR2        | R/W | H'0000        | H'FFFFC8AC | 8, 16, 32   |
| Port L control register 1 | PLCR1        | R/W | H'0000        | H'FFFFC8AE | 8, 16       |

Notes: 1. On-chip ROM disabled extension mode (area 0: 8 bits)

- 2. On-chip ROM disabled extension mode (area 0: 16 bits)
- 3. On-chip ROM enabled extension mode
- 4. Single-chip mode

#### 22.1.1 Port A I/O Register (PAIOR)

PAIOR is a 16-bit readable/writable register that sets the I/O direction of the port A pins.

PAIOR is enabled only when the port A pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PAIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | 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 | PA1<br>IOR | PA0<br>IOR |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

|         |                      | Initial |     |  |
|---------|----------------------|---------|-----|--|
| Bit     | Bit Name             | Value   | R/W | Description  |
| 15 to 0 | PA15IOR to<br>PA0IOR | O IIA o | R/W | The PA15IOR to PA0IOR bits correspond to the PA15 to PA0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|         |                      |         |     | 0: The corresponding pin is set to input.  |
|         |                      |         |     | 1: The corresponding pin is set to output.   |

## 22.1.2 Port A Control Registers 1 to 4 (PACR1 to PACR4)

PACR1 to PACR4 are 16-bit readable/writable registers that control the functions of multiplexed pins of port A. PACR1 to PACR4 are initialized to the values shown in table 22.13 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

Table 22.13 Initial Values of Port A Control Registers

|               | Initial Values                      |  |                  |  |  |  |  |  |  |  |
|---------------|-------------------------------------|--|------------------|--|--|--|--|--|--|--|
| Register Name | On-Chip ROM Disabled Extension Mode | On-Chip ROM<br>Enabled Extension<br>Mode | Single-Chip Mode |  |  |  |  |  |  |  |
| PACR4         | H'1111                              | H'1111                                   | H'0000           |  |  |  |  |  |  |  |
| PACR3         | H'1111                              | H'1111                                   | H'0000           |  |  |  |  |  |  |  |
| PACR2         | H'1111                              | H'1111                                   | H'0000           |  |  |  |  |  |  |  |
| PACR1         | H'1111                              | H'1111                                   | H'0000           |  |  |  |  |  |  |  |

# (1) Port A Control Register 4 (PACR4)

| Bit:           | 15 | 14 | 13 | 12         | 11 | 10 | 9 | 8          | 7 | 6 | 5 | 4          | 3 | 2 | 1 | 0          |
|----------------|----|----|----|------------|----|----|---|------------|---|---|---|------------|---|---|---|------------|
|                | -  | -  | -  | PA15<br>MD | -  | -  | - | PA14<br>MD | - | - | - | PA13<br>MD | - | - | - | PA12<br>MD |
| Initial value: | 0  | 0  | 0  | 0/1*       | 0  | 0  | 0 | 0/1*       | 0 | 0 | 0 | 0/1*       | 0 | 0 | 0 | 0/1*       |
| R/W:           | R  | R  | R  | R/W        | R  | R  | R | R/W        | R | R | R | R/W        | R | R | R | R/W        |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 15 to 13 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 12       | PA15MD   | 0/1*             | R/W | PA15 Mode  |
|          |          |                  |     | Controls the function of the PA15/A15 pin.                           |
|          |          |                  |     | On-chip ROM disabled extension mode                                  |
|          |          |                  |     | 0: A15 output (BSC)  |
|          |          |                  |     | 1: A15 output (BSC) (initial value)                                  |
|          |          |                  |     | On-chip ROM enabled extension mode                                   |
|          |          |                  |     | 0: PA15 input/output (port)  |
|          |          |                  |     | 1: A15 output (BSC) (initial value)                                  |
|          |          |                  |     | Single-chip mode   |
|          |          |                  |     | 0: PA15 input/output (port) (initial value)                          |
|          |          |                  |     | 1: PA15 input/output (port)  |
| 11 to 9  |          | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 8      | PA14MD   | 0/1*             | R/W | PA14 Mode  |
|        |          |                  |     | Controls the function of the PA14/A14 pin.                           |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode</li> </ul>              |
|        |          |                  |     | 0: A14 output (BSC)  |
|        |          |                  |     | 1: A14 output (BSC) (initial value)                                  |
|        |          |                  |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>               |
|        |          |                  |     | 0: PA14 input/output (port)  |
|        |          |                  |     | 1: A14 output (BSC) (initial value)                                  |
|        |          |                  |     | Single-chip mode   |
|        |          |                  |     | 0: PA14 input/output (port) (initial value)                          |
|        |          |                  |     | 1: PA14 input/output (port)  |
| 7 to 5 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 4      | PA13MD   | 0/1*             | R/W | PA13 Mode  |
|        |          |                  |     | Controls the function of the PA13/A13 pin.                           |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode</li> </ul>              |
|        |          |                  |     | 0: A13 output (BSC)  |
|        |          |                  |     | 1: A13 output (BSC) (initial value)                                  |
|        |          |                  |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>               |
|        |          |                  |     | 0: PA13 input/output (port)  |
|        |          |                  |     | 1: A13 output (BSC) (initial value)                                  |
|        |          |                  |     | Single-chip mode   |
|        |          |                  |     | 0: PA13 input/output (port) (initial value)                          |
|        |          |                  |     | 1: PA13 input/output (port)  |
| 3 to 1 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial<br>Value | R/W | Description                                 |
|-----|----------|------------------|-----|---|
| 0   | PA12MD   | 0/1*             | R/W | PA12 Mode                                   |
|     |          |                  |     | Controls the function of the PA12/A12 pin.  |
|     |          |                  |     | On-chip ROM disabled extension mode         |
|     |          |                  |     | 0: A12 output (BSC)                         |
|     |          |                  |     | 1: A12 output (BSC) (initial value)         |
|     |          |                  |     | On-chip ROM enabled extension mode          |
|     |          |                  |     | 0: PA12 input/output (port)                 |
|     |          |                  |     | 1: A12 output (BSC) (initial value)         |
|     |          |                  |     | Single-chip mode                            |
|     |          |                  |     | 0: PA12 input/output (port) (initial value) |
|     |          |                  |     | 1: PA12 input/output (port)                 |

# (2) Port A Control Register 3 (PACR3)

| Bit:           | 15 | 14 | 13 | 12         | 11 | 10 | 9 | 8          | 7 | 6 | 5 | 4         | 3 | 2 | 1 | 0         |
|----------------|----|----|----|------------|----|----|---|------------|---|---|---|-----------|---|---|---|-----------|
|                | -  | -  | -  | PA11<br>MD | -  | -  | - | PA10<br>MD | - | - | - | PA9<br>MD | - | - | - | PA8<br>MD |
| Initial value: | 0  | 0  | 0  | 0/1*       | 0  | 0  | 0 | 0/1*       | 0 | 0 | 0 | 0/1*      | 0 | 0 | 0 | 0/1*      |
| R/W:           | R  | R  | R  | R/W        | R  | R  | R | R/W        | R | R | R | R/W       | R | R | R | R/W       |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 15 to 13 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 12       | PA11MD   | 0/1*             | R/W | PA11 Mode  |
|          |          |                  |     | Controls the function of the PA11/A11 pin.                           |
|          |          |                  |     | On-chip ROM disabled extension mode                                  |
|          |          |                  |     | 0: A11 output (BSC)  |
|          |          |                  |     | 1: A11 output (BSC) (initial value)                                  |
|          |          |                  |     | On-chip ROM enabled extension mode                                   |
|          |          |                  |     | 0: PA11 input/output (port)  |
|          |          |                  |     | 1: A11 output (BSC) (initial value)                                  |
|          |          |                  |     | Single-chip mode   |
|          |          |                  |     | 0: PA11 input/output (port) (initial value)                          |
|          |          |                  |     | 1: PA11 input/output (port)  |
| 11 to 9  | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit    | Bit Name | Initial<br>Value  | R/W        | Description   |
|--------|----------|-------------------|------------|---|
| 8<br>8 | PA10MD   | <b>Value</b> 0/1* | R/W<br>R/W | PA10 Mode Controls the function of the PA10/A10 pin.  On-chip ROM disabled extension mode 0: A10 output (BSC) 1: A10 output (BSC) (initial value)  On-chip ROM enabled extension mode 0: PA10 input/output (port) 1: A10 output (BSC) (initial value)  Single-chip mode 0: PA10 input/output (port) (initial value)                           |
| 7 to 5 | _        | All 0             | R          | 1: PA10 input/output (port)  Reserved  These bits are always read as 0. The write value should always be 0.   |
| 4      | PA9MD    | 0/1*              | R/W        | PA9 Mode  Controls the function of the PA9/A9 pin.  On-chip ROM disabled extension mode  0: A9 output (BSC)  1: A9 output (BSC) (initial value)  On-chip ROM enabled extension mode  0: PA9 input/output (port)  1: A9 output (BSC) (initial value)  Single-chip mode  0: PA9 input/output (port) (initial value)  1: PA9 input/output (port) |
| 3 to 1 | _        | All 0             | R          | Reserved These bits are always read as 0. The write value should always be 0.   |

|     |          | Initial |     |  |
|-----|----------|---------|-----|--|
| Bit | Bit Name | Value   | R/W | Description                                |
| 0   | PA8MD    | 0/1*    | R/W | PA8 Mode                                   |
|     |          |         |     | Controls the function of the PA8/A8 pin.   |
|     |          |         |     | On-chip ROM disabled extension mode        |
|     |          |         |     | 0: A8 output (BSC)                         |
|     |          |         |     | 1: A8 output (BSC) (initial value)         |
|     |          |         |     | On-chip ROM enabled extension mode         |
|     |          |         |     | 0: PA8 input/output (port)                 |
|     |          |         |     | 1: A8 output (BSC) (initial value)         |
|     |          |         |     | Single-chip mode                           |
|     |          |         |     | 0: PA8 input/output (port) (initial value) |
|     |          |         |     | 1: PA8 input/output (port)                 |

# (3) Port A Control Register 2 (PACR2)

| Bit:           | 15 | 14 | 13 | 12        | 11 | 10 | 9 | 8         | 7 | 6 | 5 | 4         | 3 | 2 | 1 | 0         |
|----------------|----|----|----|-----------|----|----|---|-----------|---|---|---|-----------|---|---|---|-----------|
|                | -  | -  | -  | PA7<br>MD | -  | -  | - | PA6<br>MD | - | - | - | PA5<br>MD | - | - | - | PA4<br>MD |
| Initial value: | 0  | 0  | 0  | 0/1*      | 0  | 0  | 0 | 0/1*      | 0 | 0 | 0 | 0/1*      | 0 | 0 | 0 | 0/1*      |
| R/W:           | R  | R  | R  | R/W       | R  | R  | R | R/W       | R | R | R | R/W       | R | R | R | R/W       |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 15 to 13 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 12       | PA7MD    | 0/1*             | R/W | PA7 Mode   |
|          |          |                  |     | Controls the function of the PA7/A7 pin.                             |
|          |          |                  |     | On-chip ROM disabled extension mode                                  |
|          |          |                  |     | 0: A7 output (BSC)   |
|          |          |                  |     | 1: A7 output (BSC) (initial value)                                   |
|          |          |                  |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>               |
|          |          |                  |     | 0: PA7 input/output (port)   |
|          |          |                  |     | 1: A7 output (BSC) (initial value)                                   |
|          |          |                  |     | Single-chip mode   |
|          |          |                  |     | 0: PA7 input/output (port) (initial value)                           |
|          |          |                  |     | 1: PA7 input/output (port)   |
| 11 to 9  | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 8      | PA6MD    | 0/1*             | R/W | PA6 Mode   |
|        |          |                  |     | Controls the function of the PA6/A6 pin.                             |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode</li> </ul>              |
|        |          |                  |     | 0: A6 output (BSC)   |
|        |          |                  |     | 1: A6 output (BSC) (initial value)                                   |
|        |          |                  |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>               |
|        |          |                  |     | 0: PA6 input/output (port)   |
|        |          |                  |     | 1: A6 output (BSC) (initial value)                                   |
|        |          |                  |     | Single-chip mode   |
|        |          |                  |     | 0: PA6 input/output (port) (initial value)                           |
|        |          |                  |     | 1: PA6 input/output (port)   |
| 7 to 5 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 4      | PA5MD    | 0/1*             | R/W | PA5 Mode   |
|        |          |                  |     | Controls the function of the PA5/A5 pin.                             |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode</li> </ul>              |
|        |          |                  |     | 0: A5 output (BSC)   |
|        |          |                  |     | 1: A5 output (BSC) (initial value)                                   |
|        |          |                  |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>               |
|        |          |                  |     | 0: PA5 input/output (port)   |
|        |          |                  |     | 1: A5 output (BSC) (initial value)                                   |
|        |          |                  |     | Single-chip mode   |
|        |          |                  |     | 0: PA5 input/output (port) (initial value)                           |
|        |          |                  |     | 1: PA5 input/output (port)   |
| 3 to 1 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 0   | PA4MD    | 0/1*             | R/W | PA4 Mode  |
|     |          |                  |     | Controls the function of the PA4/A4 pin.                |
|     |          |                  |     | <ul> <li>On-chip ROM disabled extension mode</li> </ul> |
|     |          |                  |     | 0: A4 output (BSC)                                      |
|     |          |                  |     | 1: A4 output (BSC) (initial value)                      |
|     |          |                  |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>  |
|     |          |                  |     | 0: PA4 input/output (port)                              |
|     |          |                  |     | 1: A4 output (BSC) (initial value)                      |
|     |          |                  |     | Single-chip mode  |
|     |          |                  |     | 0: PA4 input/output (port) (initial value)              |
|     |          |                  |     | 1: PA4 input/output (port)                              |

# (4) Port A Control Register 1 (PACR1)

| Bit:           | 15 | 14 | 13 | 12        | 11 | 10 | 9 | 8         | 7 | 6 | 5 | 4         | 3 | 2 | 1 | 0         |
|----------------|----|----|----|-----------|----|----|---|-----------|---|---|---|-----------|---|---|---|-----------|
|                | -  | -  | -  | PA3<br>MD | -  | -  | - | PA2<br>MD | - | - | - | PA1<br>MD | - | - | - | PA0<br>MD |
| Initial value: | 0  | 0  | 0  | 0/1*      | 0  | 0  | 0 | 0/1*      | 0 | 0 | 0 | 0/1*      | 0 | 0 | 0 | 0/1*      |
| R/W:           | R  | R  | R  | R/W       | R  | R  | R | R/W       | R | R | R | R/W       | R | R | R | R/W       |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 15 to 13 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 12       | PA3MD    | 0/1*             | R/W | PA3 Mode   |
|          |          |                  |     | Controls the function of the PA3/A3 pin.                             |
|          |          |                  |     | On-chip ROM disabled extension mode                                  |
|          |          |                  |     | 0: A3 output (BSC)   |
|          |          |                  |     | 1: A3 output (BSC) (initial value)                                   |
|          |          |                  |     | On-chip ROM enabled extension mode                                   |
|          |          |                  |     | 0: PA3 input/output (port)   |
|          |          |                  |     | 1: A3 output (BSC) (initial value)                                   |
|          |          |                  |     | Single-chip mode   |
|          |          |                  |     | 0: PA3 input/output (port) (initial value)                           |
|          |          |                  |     | 1: PA3 input/output (port)   |
| 11 to 9  | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 8      | PA2MD    | 0/1*             | R/W | PA2 Mode   |
|        |          |                  |     | Controls the function of the PA2/A2 pin.                             |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode</li> </ul>              |
|        |          |                  |     | 0: A2 output (BSC)   |
|        |          |                  |     | 1: A2 output (BSC) (initial value)                                   |
|        |          |                  |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>               |
|        |          |                  |     | 0: PA2 input/output (port)   |
|        |          |                  |     | 1: A2 output (BSC) (initial value)                                   |
|        |          |                  |     | Single-chip mode   |
|        |          |                  |     | 0: PA2 input/output (port) (initial value)                           |
|        |          |                  |     | 1: PA2 input/output (port)   |
| 7 to 5 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 4      | PA1MD    | 0/1*             | R/W | PA1 Mode   |
|        |          |                  |     | Controls the function of the PA1/A1 pin.                             |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode</li> </ul>              |
|        |          |                  |     | 0: A1 output (BSC)   |
|        |          |                  |     | 1: A1 output (BSC) (initial value)                                   |
|        |          |                  |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>               |
|        |          |                  |     | 0: PA1 input/output (port)   |
|        |          |                  |     | 1: A1 output (BSC) (initial value)                                   |
|        |          |                  |     | Single-chip mode   |
|        |          |                  |     | 0: PA1 input/output (port) (initial value)                           |
|        |          |                  |     | 1: PA1 input/output (port)   |
| 3 to 1 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |

|     |          | Initial |     |  |
|-----|----------|---------|-----|--|
| Bit | Bit Name | Value   | R/W | Description                                |
| 0   | PA0MD    | 0/1*    | R/W | PA0 Mode                                   |
|     |          |         |     | Controls the function of the PA0/A0 pin.   |
|     |          |         |     | On-chip ROM disabled extension mode        |
|     |          |         |     | 0: A0 output (BSC)                         |
|     |          |         |     | 1: A0 output (BSC) (initial value)         |
|     |          |         |     | On-chip ROM enabled extension mode         |
|     |          |         |     | 0: PA0 input/output (port)                 |
|     |          |         |     | 1: A0 output (BSC) (initial value)         |
|     |          |         |     | Single-chip mode                           |
|     |          |         |     | 0: PA0 input/output (port) (initial value) |
|     |          |         |     | 1: PA0 input/output (port)                 |

#### 22.1.3 Port B I/O Register (PBIOR)

PBIOR is a 16-bit readable/writable register that sets the I/O direction of the port B pins.

PBIOR is enabled only when the port B pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PBIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15 | 14          | 13          | 12          | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|----|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | -  | PB14<br>IOR | PB13<br>IOR | PB12<br>IOR | PB11<br>IOR | PB10<br>IOR | PB9<br>IOR | PB8<br>IOR | PB7<br>IOR | PB6<br>IOR | PB5<br>IOR | PB4<br>IOR | PB3<br>IOR | PB2<br>IOR | PB1<br>IOR | PB0<br>IOR |
| Initial value: | 0  | 0           | 0           | 0           | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R  | R/W         | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

| Bit     | Bit Name             | Initial<br>Value | R/W | Description  |
|---------|----------------------|------------------|-----|--|
| 15      | _                    | 0                | R   | Reserved   |
|         |                      |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 14 to 0 | PB14IOR to<br>PB0IOR | All 0            | R/W | The PB14IOR to PB0IOR bits correspond to the PB14 to PB0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|         |                      |                  |     | 0: The corresponding pin is set to input.  |
|         |                      |                  |     | 1: The corresponding pin is set to output.   |

#### 22.1.4 Port B Control Registers 1 to 4 (PBCR1 to PBCR4)

PBCR1 to PBCR4 are 16-bit readable/writable registers that control the functions of multiplexed pins of port B.

PBCR1 to PBCR4 are initialized to the values shown in table 22.14 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

Table 22.14 Initial Values of Port B Control Registers

|               |                | Ini                      | tial Values                      |                  |  |  |  |  |  |
|---------------|----------------|--------------------------|----------------------------------|------------------|--|--|--|--|--|
|               |                | hip ROM<br>ktension Mode | On-Chip ROM<br>Enabled Extension |                  |  |  |  |  |  |
| Register Name | Area 0: 8 Bits | Area 0: 16 Bits          | Mode                             | Single-Chip Mode |  |  |  |  |  |
| PBCR4         | H'0400         | H'0400                   | H'0400                           | H'0000           |  |  |  |  |  |
| PBCR3         | H'0110         | H'0110                   | H'0112                           | H'0000           |  |  |  |  |  |
| PBCR2         | H'0122         | H'1122                   | H'1122                           | H'0000           |  |  |  |  |  |
| PBCR1         | H'2222         | H'2222                   | H'2222                           | H'0000           |  |  |  |  |  |

## (1) Port B Control Register 4 (PBCR4)

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10   | 9       | 8    | 7 | 6 | 5     | 4        | 3 | 2 | 1     | 0        |
|----------------|----|----|----|----|----|------|---------|------|---|---|-------|----------|---|---|-------|----------|
|                | -  | -  | -  | -  | -  | PE   | 314MD[2 | ::0] | - | - | PB13N | /ID[1:0] | - | - | PB12N | /ID[1:0] |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0/1* | 0       | 0    | 0 | 0 | 0     | 0        | 0 | 0 | 0     | 0        |
| R/W:           | R  | R  | R  | R  | R  | R/W  | R/W     | R/W  | R | R | R/W   | R/W      | R | R | R/W   | R/W      |

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

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 15 to 11 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 10 to 8 | PB14MD   | 000/100*         | R/W | PB14 Mode  |
|         | [2:0]    |                  |     | Control the function of the PB14/RD/WR pin.                          |
|         |          |                  |     | On-chip ROM disabled extension mode                                  |
|         |          |                  |     | 000: RD/WR output (BSC)  |
|         |          |                  |     | 001: Setting prohibited  |
|         |          |                  |     | 010: Setting prohibited  |
|         |          |                  |     | 011: Setting prohibited  |
|         |          |                  |     | 100: RD/WR output (BSC) (initial value)                              |
|         |          |                  |     | 101: Setting prohibited  |
|         |          |                  |     | 110: Setting prohibited  |
|         |          |                  |     | 111: Setting prohibited  |
|         |          |                  |     | On-chip ROM enabled extension mode                                   |
|         |          |                  |     | 000: PB14 input/output (port)  |
|         |          |                  |     | 001: Setting prohibited  |
|         |          |                  |     | 010: Setting prohibited  |
|         |          |                  |     | 011: Setting prohibited  |
|         |          |                  |     | 100: RD/WR output (BSC) (initial value)                              |
|         |          |                  |     | 101: Setting prohibited  |
|         |          |                  |     | 110: Setting prohibited  |
|         |          |                  |     | 111: Setting prohibited  |
|         |          |                  |     | Single-chip mode   |
|         |          |                  |     | 000: PB14 input/output (port) (initial value)                        |
|         |          |                  |     | 001: Setting prohibited  |
|         |          |                  |     | 010: Setting prohibited  |
|         |          |                  |     | 011: Setting prohibited  |
|         |          |                  |     | 100: PB14 input/output (port)  |
|         |          |                  |     | 101: Setting prohibited  |
|         |          |                  |     | 110: Setting prohibited  |
|         |          |                  |     | 111: Setting prohibited  |
| 7, 6    | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 5, 4 | PB13MD   | 00               | R/W | PB13 Mode  |
|      | [1:0]    |                  |     | Control the function of the PB13/CS3/RSPCKB pin.                     |
|      |          |                  |     | Extension mode   |
|      |          |                  |     | 00: PB13 input/output (port) (initial value)                         |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: CS3 output (BSC)   |
|      |          |                  |     | 11: RSPCKB input/output (RSPI)                                       |
|      |          |                  |     | Single-chip mode   |
|      |          |                  |     | 00: PB13 input/output (port) (initial value)                         |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: PB13 input/output (port)   |
|      |          |                  |     | 11: RSPCKB input/output (RSPI)                                       |
| 3, 2 | _        | All 0            | R   | Reserved   |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 1, 0 | PB12MD   | 00               | R/W | PA12 Mode  |
|      | [1:0]    |                  |     | Control the function of the PB12/CS2/RSPCKA pin.                     |
|      |          |                  |     | Extension mode   |
|      |          |                  |     | 00: PB12 input/output (port) (initial value)                         |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: CS2 output (BSC)   |
|      |          |                  |     | 11: RSPCKA input/output (RSPI)                                       |
|      |          |                  |     | Single-chip mode   |
|      |          |                  |     | 00: PB12 input/output (port) (initial value)                         |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: PB12 input/output (port)   |
|      |          |                  |     | 11: RSPCKA input/output (RSPI)                                       |

# (2) Port B Control Register 3 (PBCR3)

| Bit:           | 15 | 14 | 13    | 12      | 11 | 10 | 9 | 8          | 7 | 6 | 5 | 4         | 3 | 2 | 1    | 0      |
|----------------|----|----|-------|---------|----|----|---|------------|---|---|---|-----------|---|---|------|--------|
|                | -  | -  | PB11M | 1D[1:0] | -  | -  | - | PB10<br>MD | - | - | - | PB9<br>MD | - | - | PB8M | D[1:0] |
| Initial value: | 0  | 0  | 0     | 0       | 0  | 0  | 0 | 0/1*       | 0 | 0 | 0 | 0/1*      | 0 | 0 | 0/1* | 0      |
| R/W:           | R  | R  | R/W   | R/W     | R  | R  | R | R/W        | R | R | R | R/W       | R | R | R/W  | R/W    |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 15, 14  | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 13, 12  | PB11MD   | 00               | R/W | PB11 Mode  |
|         | [1:0]    |                  |     | Control the function of the PB11/CS1/TOE21 pin.                      |
|         |          |                  |     | Extension mode   |
|         |          |                  |     | 00: PB11 input/output (port) (initial value)                         |
|         |          |                  |     | 01: Setting prohibited   |
|         |          |                  |     | 10: CS1 output (BSC)   |
|         |          |                  |     | 11: TOE21 output (ATU-III)   |
|         |          |                  |     | Single-chip mode   |
|         |          |                  |     | 00: PB11 input/output (port) (initial value)                         |
|         |          |                  |     | 01: Setting prohibited   |
|         |          |                  |     | 10: PB11 input/output (port)   |
|         |          |                  |     | 11: TOE21 output (ATU-III)   |
| 11 to 9 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 8      | PB10MD   | 0/1*             | R/W | PB10 Mode  Controls the function of the PB10/CS0 pin.  On-chip ROM disabled extension mode  CS0 output (BSC)  CS0 output (BSC) (initial value)  On-chip ROM enabled extension mode  PB10 input/output (port)  CS0 output (BSC) (initial value)  Single-chip mode  PB10 input/output (port) (initial value)  PB10 input/output (port) (initial value)  PB10 input/output (port) |
| 7 to 5 | _        | All 0            | R   | Reserved These bits are always read as 0. The write value should always be 0.  |
| 4      | PB9MD    | 0/1*             | R/W | PB9 Mode  Controls the function of the PB9/RD pin.  On-chip ROM disabled extension mode  RD output (BSC)  RD output (BSC) (initial value)  On-chip ROM enabled extension mode  PB9 input/output (port)  RD output (BSC) (initial value)  Single-chip mode  PB9 input/output (port) (initial value)  PB9 input/output (port) (initial value)                                    |
| 3, 2   |          | All 0            | R   | Reserved These bits are always read as 0. The write value should always be 0.  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description                                     |
|------|----------|------------------|-----|---|
| 1, 0 | PB8MD    | 00/10*           | R/W | PB8 Mode  |
|      | [1:0]    |                  |     | Control the function of the PB8/WAIT/TOE20 pin. |
|      |          |                  |     | On-chip ROM disabled extension mode             |
|      |          |                  |     | 00: WAIT input (BSC) (initial value)            |
|      |          |                  |     | 01: Setting prohibited                          |
|      |          |                  |     | 10: WAIT input (BSC)                            |
|      |          |                  |     | 11: WAIT input (BSC)                            |
|      |          |                  |     | On-chip ROM enabled extension mode              |
|      |          |                  |     | 00: PB8 input/output (port)                     |
|      |          |                  |     | 01: Setting prohibited                          |
|      |          |                  |     | 10: WAIT input (BSC) (initial value)            |
|      |          |                  |     | 11: TOE20 output (ATU-III)                      |
|      |          |                  |     | Single-chip mode                                |
|      |          |                  |     | 00: PB8 input/output (port) (initial value)     |
|      |          |                  |     | 01: Setting prohibited                          |
|      |          |                  |     | 10: PB8 input/output (port) (BSC)               |
|      |          |                  |     | 11: TOE20 output (ATU-III)                      |

# (3) Port B Control Register 2 (PBCR2)

| Bit:           | 15 | 14 | 13 | 12        | 11 | 10 | 9 | 8         | 7 | 6   | 5       | 4   | 3 | 2   | 1       | 0   |
|----------------|----|----|----|-----------|----|----|---|-----------|---|-----|---------|-----|---|-----|---------|-----|
|                | -  | -  | -  | PB7<br>MD | -  | -  | - | PB6<br>MD | - | Р   | B5MD[2: | 0]  | - | Р   | B4MD[2: | 0]  |
| Initial value: | 0  | 0  | 0  | 0/1*      | 0  | 0  | 0 | 0/1*      | 0 | 0   | 0/1*    | 0   | 0 | 0   | 0/1*    | 0   |
| R/W:           | R  | R  | R  | R/W       | R  | R  | R | R/W       | R | R/W | R/W     | R/W | R | R/W | R/W     | R/W |

| Bit      | Bit Name | Initial<br>Value | R/W | Description   |
|----------|----------|------------------|-----|---|
| 15 to 13 | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.          |
| 12       | PB7MD    | 0/1*             | R/W | PB7 Mode  |
|          |          |                  |     | Controls the function of the PB7/WE1 pin.                                     |
|          |          |                  |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 8 bits)</li> </ul>  |
|          |          |                  |     | 0: PB7 input/output (port) (initial value) 1: WE1 output (BSC)                |
|          |          |                  |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 16 bits)</li> </ul> |
|          |          |                  |     | 0: WE1 output (BSC)   |
|          |          |                  |     | 1: WE1 output (BSC) (initial value)   |
|          |          |                  |     | On-chip ROM enabled extension mode  |
|          |          |                  |     | 0: PB7 input/output (port)  |
|          |          |                  |     | 1: WE1 output (BSC) (initial value)   |
|          |          |                  |     | Single-chip mode  |
|          |          |                  |     | 0: PB7 input/output (port) (initial value)                                    |
|          |          |                  |     | 1: PB7 input/output (port)  |
| 11 to 9  |          | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.          |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 8   | PB6MD    | 0/1*             | R/W | PB6 Mode  |
|     |          |                  |     | Controls the function of the PB6/WE0 pin.                         |
|     |          |                  |     | On-chip ROM disabled extension mode                               |
|     |          |                  |     | 0: WE0 output (BSC)   |
|     |          |                  |     | 1: WE0 output (BSC) (initial value)                               |
|     |          |                  |     | On-chip ROM enabled extension mode                                |
|     |          |                  |     | 0: PB6 input/output (port)  |
|     |          |                  |     | 1: WE0 output (BSC) (initial value)                               |
|     |          |                  |     | Single-chip mode  |
|     |          |                  |     | 0: PB6 input/output (port) (initial value)                        |
|     |          |                  |     | 1: PB6 input/output (port)  |
| 7   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0. |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 6 to 4 | PB5MD    | 000/010*         | R/W | PB5 Mode  |
|        | [2:0]    |                  |     | Control the function of the PB5/A21/CRx_B/TIF7 pin.               |
|        |          |                  |     | On-chip ROM disabled extension mode                               |
|        |          |                  |     | 000: A21 output (address)   |
|        |          |                  |     | 001: Setting prohibited   |
|        |          |                  |     | 010: A21 output (address) (initial value)                         |
|        |          |                  |     | 011: A21 output (address)   |
|        |          |                  |     | 100: A21 output (address)   |
|        |          |                  |     | 101: Setting prohibited   |
|        |          |                  |     | 110: Setting prohibited   |
|        |          |                  |     | 111: Setting prohibited   |
|        |          |                  |     | On-chip ROM enabled extension mode                                |
|        |          |                  |     | 000: PB5 input/output (port)                                      |
|        |          |                  |     | 001: Setting prohibited   |
|        |          |                  |     | 010: A21 output (address) (initial value)                         |
|        |          |                  |     | 011: CRx_B input (RCAN-TL1)                                       |
|        |          |                  |     | 100: TIF7 input (ATU-III)   |
|        |          |                  |     | 101: Setting prohibited   |
|        |          |                  |     | 110: Setting prohibited   |
|        |          |                  |     | 111: Setting prohibited   |
|        |          |                  |     | Single-chip mode  |
|        |          |                  |     | 000: PB5 input/output (port) (initial value)                      |
|        |          |                  |     | 001: Setting prohibited   |
|        |          |                  |     | 010: PB5 input/output (port)                                      |
|        |          |                  |     | 011: CRx_B input (RCAN-TL1)                                       |
|        |          |                  |     | 100: TIF7 input (ATU-III)   |
|        |          |                  |     | 101: Setting prohibited   |
|        |          |                  |     | 110: Setting prohibited   |
|        |          |                  |     | 111: Setting prohibited   |
| 3      | _        | 0                | R   | Reserved  |
|        |          |                  |     | This bit is always read as 0. The write value should always be 0. |

| Bit     | Bit Name | Initial<br>Value | R/W | Description   |
|---------|----------|------------------|-----|---|
| 2 to 0  | PB4MD    | 000/010*         | R/W | PB4 Mode  |
|         | [2:0]    |                  |     | Control the function of the PB4/A20/CTx_B/TIF6 pin. |
|         |          |                  |     | On-chip ROM disabled extension mode                 |
|         |          |                  |     | 000: A20 output (address)                           |
|         |          |                  |     | 001: Setting prohibited                             |
|         |          |                  |     | 010: A20 output (address) (initial value)           |
|         |          |                  |     | 011: A20 output (address)                           |
|         |          |                  |     | 100: A20 output (address)                           |
|         |          |                  |     | 101: Setting prohibited                             |
|         |          |                  |     | 110: Setting prohibited                             |
|         |          |                  |     | 111: Setting prohibited                             |
|         |          |                  |     | On-chip ROM enabled extension mode                  |
|         |          |                  |     | 000: PB4 input/output (port)                        |
|         |          |                  |     | 001: Setting prohibited                             |
|         |          |                  |     | 010: A20 output (address) (initial value)           |
|         |          |                  |     | 011: CTx_B output (RCAN-TL1)                        |
|         |          |                  |     | 100: TIF6 input (ATU-III)                           |
|         |          |                  |     | 101: Setting prohibited                             |
|         |          |                  |     | 110: Setting prohibited                             |
|         |          |                  |     | 111: Setting prohibited                             |
|         |          |                  |     | Single-chip mode                                    |
|         |          |                  |     | 000: PB4 input/output (port) (initial value)        |
|         |          |                  |     | 001: Setting prohibited                             |
|         |          |                  |     | 010: PB4 input/output (port)                        |
|         |          |                  |     | 011: CTx_B output (RCAN-TL1)                        |
|         |          |                  |     | 100: TIF6 input (ATU-III)                           |
|         |          |                  |     | 101: Setting prohibited                             |
|         |          |                  |     | 110: Setting prohibited                             |
| Noto: * |          |                  |     | 111: Setting prohibited                             |

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

# (4) Port B Control Register 1 (PBCR1)

| Bit:           | 15 | 14 | 13   | 12     | 11 | 10 | 9    | 8      | 7 | 6 | 5    | 4       | 3 | 2 | 1    | 0      |
|----------------|----|----|------|--------|----|----|------|--------|---|---|------|---------|---|---|------|--------|
|                | -  | -  | РВЗМ | D[1:0] | -  | -  | PB2M | D[1:0] | - | - | PB1M | ID[1:0] | - | - | PB0M | D[1:0] |
| Initial value: | 0  | 0  | 0/1* | 0      | 0  | 0  | 0/1* | 0      | 0 | 0 | 0/1* | 0       | 0 | 0 | 0/1* | 0      |
| R/W:           | R  | R  | R/W  | R/W    | R  | R  | R/W  | R/W    | R | R | R/W  | R/W     | R | R | R/W  | R/W    |

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

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 15, 14 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 13, 12 | PB3MD    | 00/10*           | R/W | PB3 Mode   |
|        | [1:0]    |                  |     | Control the function of the PB3/A19/MISOB pin.                       |
|        |          |                  |     | On-chip ROM disabled extension mode                                  |
|        |          |                  |     | 00: A19 output (address)   |
|        |          |                  |     | 01: Setting prohibited   |
|        |          |                  |     | 10: A19 output (address) (initial value)                             |
|        |          |                  |     | 11: A19 output (address)   |
|        |          |                  |     | On-chip ROM enabled extension mode                                   |
|        |          |                  |     | 00: PB3 input/output (port)  |
|        |          |                  |     | 01: Setting prohibited   |
|        |          |                  |     | 10: A19 output (address) (initial value)                             |
|        |          |                  |     | 11: MISOB input/output (RSPI)  |
|        |          |                  |     | Single-chip mode   |
|        |          |                  |     | 00: PB3 input/output (port) (initial value)                          |
|        |          |                  |     | 01: Setting prohibited   |
|        |          |                  |     | 10: PB3 input/output (port)  |
|        |          |                  |     | 11: MISOB input/output (RSPI)  |
| 11, 10 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 9, 8 | PB2MD    | 00/10*           | R/W | PB2 Mode   |
|      | [1:0]    |                  |     | Control the function of the PB2/A18/MOSIB pin.                       |
|      |          |                  |     | On-chip ROM disabled extension mode                                  |
|      |          |                  |     | 00: A18 output (address)   |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: A18 output (address) (initial value)                             |
|      |          |                  |     | 11: A18 output (address)   |
|      |          |                  |     | On-chip ROM enabled extension mode                                   |
|      |          |                  |     | 00: PB2 input/output (port)  |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: A18 output (address) (initial value)                             |
|      |          |                  |     | 11: MOSIB input/output (RSPI)  |
|      |          |                  |     | Single-chip mode   |
|      |          |                  |     | 00: PB2 input/output (port) (initial value)                          |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: PB2 input/output (port)  |
|      |          |                  |     | 11: MOSIB input/output (RSPI)  |
| 7, 6 | _        | All 0            | R   | Reserved   |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 5, 4 | PB1MD    | 00/10*           | R/W | PB1 Mode   |
|      | [1:0]    |                  |     | Control the function of the PB1/A17/MISOA pin.                       |
|      |          |                  |     | On-chip ROM disabled extension mode                                  |
|      |          |                  |     | 00: A17 output (address)   |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: A17 output (address) (initial value)                             |
|      |          |                  |     | 11: A17 output (address)   |
|      |          |                  |     | On-chip ROM enabled extension mode                                   |
|      |          |                  |     | 00: PB1 input/output (port)  |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: A17 output (address) (initial value)                             |
|      |          |                  |     | 11: MISOA input/output (RSPI)  |
|      |          |                  |     | Single-chip mode   |
|      |          |                  |     | 00: PB1 input/output (port) (initial value)                          |
|      |          |                  |     | 01: Setting prohibited   |
|      |          |                  |     | 10: PB1 input/output (port)  |
|      |          |                  |     | 11: MISOA input/output (RSPI)  |
| 3, 2 | _        | All 0            | R   | Reserved   |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0. |

| Bit  | Bit Name | Initial<br>Value | R/W | Description                                    |
|------|----------|------------------|-----|--|
| 1, 0 | PB0MD    | 00/10*           | R/W | PB0 Mode                                       |
|      | [1:0]    |                  |     | Control the function of the PB0/A16/MOSIA pin. |
|      |          |                  |     | On-chip ROM disabled extension mode            |
|      |          |                  |     | 00: A16 output (address)                       |
|      |          |                  |     | 01: Setting prohibited                         |
|      |          |                  |     | 10: A16 output (address) (initial value)       |
|      |          |                  |     | 11: A16 output (address)                       |
|      |          |                  |     | On-chip ROM enabled extension mode             |
|      |          |                  |     | 00: PB0 input/output (port)                    |
|      |          |                  |     | 01: Setting prohibited                         |
|      |          |                  |     | 10: A16 output (address) (initial value)       |
|      |          |                  |     | 11: MOSIA input/output (RSPI)                  |
|      |          |                  |     | Single-chip mode                               |
|      |          |                  |     | 00: PB0 input/output (port) (initial value)    |
|      |          |                  |     | 01: Setting prohibited                         |
|      |          |                  |     | 10: PB0 input/output (port)                    |
|      |          |                  |     | 11: MOSIA input/output (RSPI)                  |

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

### 22.1.5 Port C I/O Register (PCIOR)

PCIOR is a 16-bit readable/writable register that sets the I/O direction of the port C pins.

PCIOR is enabled only when the port C pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PCIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | PC15<br>IOR | PC14<br>IOR | PC13<br>IOR | PC12<br>IOR | PC11<br>IOR | PC10<br>IOR | PC9<br>IOR | PC8<br>IOR | PC7<br>IOR | PC6<br>IOR | PC5<br>IOR | PC4<br>IOR | PC3<br>IOR | PC2<br>IOR | PC1<br>IOR | PC0<br>IOR |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

|         |                         | Initial |     |  |
|---------|-------------------------|---------|-----|--|
| Bit     | Bit Name                | Value   | R/W | Description  |
| 15 to 0 | PC15IOR<br>to<br>PC0IOR | All 0   | R/W | The PC15IOR to PC0IOR bits correspond to the PC15 to PC0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|         |                         |         |     | 0: The corresponding pin is set to input.  |
|         |                         |         |     | 1: The corresponding pin is set to output.   |

### 22.1.6 Port C Control Registers 1 to 4 (PCCR1 to PCCR4)

PCCR1 to PCCR4 are 16-bit readable/writable registers that control the functions of multiplexed pins of port C.

PCCR1 to PCCR4 are initialized to the values shown in table 22.15 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

Table 22.15 Initial Values of Port C Control Registers

|               | Initial Values |                          |                                   |                  |  |  |  |  |  |  |  |
|---------------|----------------|--------------------------|-----------------------------------|------------------|--|--|--|--|--|--|--|
|               |                | hip ROM<br>xtension Mode | On-Chip ROM<br>_Enabled Extension |                  |  |  |  |  |  |  |  |
| Register Name | Area 0: 8 Bits | Area 0: 16 Bits          | Mode                              | Single-Chip Mode |  |  |  |  |  |  |  |
| PCCR4         | H'0000         | H'1111                   | H'1111                            | H'0000           |  |  |  |  |  |  |  |
| PCCR3         | H'0000         | H'1111                   | H'1111                            | H'0000           |  |  |  |  |  |  |  |
| PCCR2         | H'1111         | H'1111                   | H'1111                            | H'0000           |  |  |  |  |  |  |  |
| PCCR1         | H'1111         | H'1111                   | H'1111                            | H'0000           |  |  |  |  |  |  |  |

# (1) Port C Control Register 4 (PCCR4)

| Bit:           | 15 | 14 | 13 | 12         | 11 | 10 | 9 | 8          | 7 | 6 | 5 | 4          | 3 | 2 | 1 | 0          |
|----------------|----|----|----|------------|----|----|---|------------|---|---|---|------------|---|---|---|------------|
|                | -  | -  | -  | PC15<br>MD | -  | -  | - | PC14<br>MD | - | - | - | PC13<br>MD | - | - | - | PC12<br>MD |
| Initial value: | 0  | 0  | 0  | 0/1*1      | 0  | 0  | 0 | 0/1*1      | 0 | 0 | 0 | 0/1*1      | 0 | 0 | 0 | 0/1*1      |
| R/W:           | R  | R  | R  | R/W        | R  | R  | R | R/W        | R | R | R | R/W        | R | R | R | R/W        |

Note: 1. The initial value depends on the operating mode of the LSI.

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 15 to 13 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.         |
| 12       | PC15MD   | 0/1*1            | R/W | PC15 Mode  |
|          |          |                  |     | Controls the function of the PC15/D15 pin.                                   |
|          |          |                  |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 8 bits)</li> </ul> |
|          |          |                  |     | 0: PC15 input/output (port) (initial value)                                  |
|          |          |                  |     | 1*2: D15 input/output (BSC)  |
|          |          |                  |     | <ul> <li>On-chip ROM disabled extension mode</li> </ul>                      |
|          |          |                  |     | (area 0: 16 bits)  |
|          |          |                  |     | 0*2: D15 input/output (BSC)  |
|          |          |                  |     | 1*2: D15 input/output (BSC) (initial value)                                  |
|          |          |                  |     | On-chip ROM enabled extension mode   |
|          |          |                  |     | 0: PC15 input/output (port)  |
|          |          |                  |     | 1*2: D15 input/output (BSC) (initial value)                                  |
|          |          |                  |     | Single-chip mode   |
|          |          |                  |     | 0: PC15 input/output (port) (initial value)                                  |
|          |          |                  |     | 1: PC15 input/output (port)  |
| 11 to 9  |          | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.         |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 8      | PC14MD   | 0/1*1            | R/W | PC14 Mode   |
|        |          |                  |     | Controls the function of the PC14/D14 pin.  |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode (area 0: 8 bits)</li> <li>PC14 input/output (port) (initial value)         1*²: D14 input/output (BSC)</li> <li>On-chip ROM disabled extension mode (area 0: 16 bits)         0*²: D14 input/output (BSC)         1*²: D14 input/output (BSC) (initial value)</li> <li>On-chip ROM enabled extension mode         0: PC14 input/output (port)         1*²: D14 input/output (BSC) (initial value)</li> <li>Single-chip mode</li> </ul> |
|        |          |                  |     | 0: PC14 input/output (port) (initial value)   |
| 7 to 5 |          | All 0            | R   | 1: PC14 input/output (port)  Reserved   |
|        |          | All V            | 11  | These bits are always read as 0. The write value should always be 0.  |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 4      | PC13MD   | 0/1*1            | R/W | PC13 Mode   |
|        |          |                  |     | Controls the function of the PC13/D13 pin.                                    |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 8 bits)</li> </ul>  |
|        |          |                  |     | 0: PC13 input/output (port) (initial value)  1*2: D13 input/output (BSC)      |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 16 bits)</li> </ul> |
|        |          |                  |     | 0*2: D13 input/output (BSC)   |
|        |          |                  |     | 1*2: D13 input/output (BSC) (initial value)                                   |
|        |          |                  |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>                        |
|        |          |                  |     | 0: PC13 input/output (port)   |
|        |          |                  |     | 1*2: D13 input/output (BSC) (initial value)                                   |
|        |          |                  |     | Single-chip mode  |
|        |          |                  |     | 0: PC13 input/output (port) (initial value)                                   |
|        |          |                  |     | 1: PC13 input/output (port)   |
| 3 to 1 | _        | All 0            | R   | Reserved  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.          |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |  |  |  |  |  |
|-----|----------|------------------|-----|---|--|--|--|--|--|
| 0   | PC12MD   | 0/1*1            | R/W | PC12 Mode   |  |  |  |  |  |
|     |          |                  |     | Controls the function of the PC12/D12 pin.                                    |  |  |  |  |  |
|     |          |                  |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 8 bits)</li> </ul>  |  |  |  |  |  |
|     |          |                  |     | 0: PC12 input/output (port) (initial value)  1*2: D12 input/output (BSC)      |  |  |  |  |  |
|     |          |                  |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 16 bits)</li> </ul> |  |  |  |  |  |
|     |          |                  |     | 0*2: D12 input/output (BSC)   |  |  |  |  |  |
|     |          |                  |     | 1*2: D12 input/output (BSC) (initial value)                                   |  |  |  |  |  |
|     |          |                  |     | On-chip ROM enabled extension mode  |  |  |  |  |  |
|     |          |                  |     | 0: PC12 input/output (port)   |  |  |  |  |  |
|     |          |                  |     | 1*2: D12 input/output (BSC) (initial value)                                   |  |  |  |  |  |
|     |          |                  |     | Single-chip mode  |  |  |  |  |  |
|     |          |                  |     | 0: PC12 input/output (port) (initial value)                                   |  |  |  |  |  |
|     |          |                  |     | 1: PC12 input/output (port)   |  |  |  |  |  |

Notes: 1. The initial value depends on the operating mode of the LSI.

2. The pin is pulled up.

# (2) Port C Control Register 3 (PCCR3)

| Bit:           | 15 | 14 | 13 | 12         | 11 | 10 | 9 | 8          | 7 | 6 | 5 | 4         | 3 | 2 | 1 | 0         |
|----------------|----|----|----|------------|----|----|---|------------|---|---|---|-----------|---|---|---|-----------|
|                | -  | -  | -  | PC11<br>MD | -  | -  | - | PC10<br>MD | - | - | - | PC9<br>MD | - | - | - | PC8<br>MD |
| Initial value: | 0  | 0  | 0  | 0/1*1      | 0  | 0  | 0 | 0/1*1      | 0 | 0 | 0 | 0/1*1     | 0 | 0 | 0 | 0/1*1     |
| R/W:           | R  | R  | R  | R/W        | R  | R  | R | R/W        | R | R | R | R/W       | R | R | R | R/W       |

Note: 1. The initial value depends on the operating mode of the LSI.

| Bit      | Bit Name | Initial<br>Value | R/W | Description   |  |  |  |  |
|----------|----------|------------------|-----|---|--|--|--|--|
| 15 to 13 | _        | All 0            | R   | Reserved  |  |  |  |  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.          |  |  |  |  |
| 12       | PC11MD   | 0/1*1            | R/W | PC11 Mode   |  |  |  |  |
|          |          |                  |     | Controls the function of the PC11/D11 pin.                                    |  |  |  |  |
|          |          |                  |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 8 bits)</li> </ul>  |  |  |  |  |
|          |          |                  |     | 0: PC11 input/output (port) (initial value)                                   |  |  |  |  |
|          |          |                  |     | 1*2: D11 input/output (BSC)   |  |  |  |  |
|          |          |                  |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 16 bits)</li> </ul> |  |  |  |  |
|          |          |                  |     | 0*2: D11 input/output (BSC)   |  |  |  |  |
|          |          |                  |     | 1*2: D11 input/output (BSC) (initial value)                                   |  |  |  |  |
|          |          |                  |     | On-chip ROM enabled extension mode  |  |  |  |  |
|          |          |                  |     | 0: PC11 input/output (port)   |  |  |  |  |
|          |          |                  |     | 1*2: D11 input/output (BSC) (initial value)                                   |  |  |  |  |
|          |          |                  |     | Single-chip mode  |  |  |  |  |
|          |          |                  |     | 0: PC11 input/output (port) (initial value)                                   |  |  |  |  |
|          |          |                  |     | 1: PC11input/output (port)  |  |  |  |  |
| 11 to 9  | _        | All 0            | R   | Reserved  |  |  |  |  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.          |  |  |  |  |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 8      | PC10MD   | 0/1*1            | R/W | PC10 Mode  |
|        |          |                  |     | Controls the function of the PC10/D10 pin.   |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode (area 0: 8 bits)</li> <li>0: PC10 input/output (port) (initial value)         1*²: D10 input/output (BSC)</li> <li>On-chip ROM disabled extension mode (area 0: 16 bits)         0*²: D10 input/output (BSC)         1*²: D10 input/output (BSC) (initial value)</li> <li>On-chip ROM enabled extension mode         0: PC10 input/output (port)         1*²: D10 input/output (BSC) (initial value)</li> <li>Single-chip mode         0: PC10 input/output (port) (initial value)</li> </ul> |
|        |          |                  |     | 1: PC10 input/output (port)  |
| 7 to 5 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 4      | PC9MD    | 0/1*1            | R/W | PC9 Mode Controls the function of the PC9/D9 pin.  |
|        |          |                  |     | <ul> <li>On-chip ROM disabled extension mode (area 0: 8 bits)</li> <li>0: PC9 input/output (port) (initial value)         1*²: D9 input/output (BSC)</li> <li>On-chip ROM disabled extension mode (area 0: 16 bits)         0*²: D9 input/output (BSC)         1*²: D9 input/output (BSC) (initial value)</li> <li>On-chip ROM enabled extension mode         0: PC9 input/output (port)         1*²: D9 input/output (BSC) (initial value)</li> <li>Single-chip mode</li> </ul> |
|        |          |                  |     | <ul><li>0: PC9 input/output (port) (initial value)</li><li>1: PC9 input/output (port)</li></ul>  |
| 3 to 1 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.   |

| Bit | Bit Name | Initial<br>Value           | R/W | Description   |
|-----|----------|----------------------------|-----|---|
| 0   | PC8MD    | 0/ <b>1</b> * <sup>1</sup> | R/W | PC8 Mode  |
|     |          |                            |     | Controls the function of the PC8/D8 pin.                                      |
|     |          |                            |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 8 bits)</li> </ul>  |
|     |          |                            |     | 0: PC8 input/output (port) (initial value)  1*2: D8 input/output (BSC)        |
|     |          |                            |     | <ul> <li>On-chip ROM disabled extension mode<br/>(area 0: 16 bits)</li> </ul> |
|     |          |                            |     | 0*2: D8 input/output (BSC)  |
|     |          |                            |     | 1*2: D8 input/output (BSC) (initial value)                                    |
|     |          |                            |     | <ul> <li>On-chip ROM enabled extension mode</li> </ul>                        |
|     |          |                            |     | 0: PC8 input/output (port)  |
|     |          |                            |     | 1*2: D8 input/output (BSC) (initial value)                                    |
|     |          |                            |     | Single-chip mode  |
|     |          |                            |     | 0: PC8 input/output (port) (initial value)                                    |
|     |          |                            |     | 1: PC8 input/output (port)  |

Notes: 1. The initial value depends on the operating mode of the LSI.

2. The pin is pulled up.

# (3) Port C Control Register 2 (PCCR2)

| Bit:           | 15 | 14 | 13 | 12        | 11 | 10 | 9 | 8         | 7 | 6 | 5 | 4         | 3 | 2 | 1 | 0         |
|----------------|----|----|----|-----------|----|----|---|-----------|---|---|---|-----------|---|---|---|-----------|
|                | -  | -  | -  | PC7<br>MD | -  | -  | - | PC6<br>MD | - | - | - | PC5<br>MD | - | - | - | PC4<br>MD |
| Initial value: | 0  | 0  | 0  | 0/1*1     | 0  | 0  | 0 | 0/1*1     | 0 | 0 | 0 | 0/1*1     | 0 | 0 | 0 | 0/1*1     |
| R/W:           | R  | R  | R  | R/W       | R  | R  | R | R/W       | R | R | R | R/W       | R | R | R | R/W       |

Note: 1. The initial value depends on the operating mode of the LSI.

| Bit      | Bit Name | Initial<br>Value | R/W | Description   |
|----------|----------|------------------|-----|---|
| 15 to 13 | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 12       | PC7MD    | 0/1*1            | R/W | PC7 Mode  |
|          |          |                  |     | Controls the function of the PC7/D7 pin.  |
|          |          |                  |     | <ul> <li>On-chip ROM disabled extension mode         0*²: D7 input/output (BSC)         1*²: D7 input/output (BSC) (initial value)</li> <li>On-chip ROM enabled extension mode         0: PC7 input/output (port)         1*²: D7 input/output (BSC) (initial value)</li> <li>Single-chip mode         0: PC7 input/output (port) (initial value)         1: PC7 input/output (port)</li> </ul> |
| 11 to 9  | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.  |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |  |  |  |  |
|--------|----------|------------------|-----|---|--|--|--|--|
| 8      | PC6MD    | 0/1*1            | R/W | PC6 Mode  Controls the function of the PC6/D6 pin.  On-chip ROM disabled extension mode  0*2: D6 input/output (BSC)  1*2: D6 input/output (BSC) (initial value)  On-chip ROM enabled extension mode  0: PC6 input/output (port)  1*2: D6 input/output (BSC) (initial value)  Single-chip mode  0: PC6 input/output (port) (initial value)  1: PC6 input/output (port) |  |  |  |  |
| 7 to 5 | _        | All 0            | R   | Reserved These bits are always read as 0. The write value should always be 0.   |  |  |  |  |
| 4      | PC5MD    | 0/1*1            | R/W | PC5 Mode  Controls the function of the PC5/D5 pin.  On-chip ROM disabled extension mode  0*2: D5 input/output (BSC)  1*2: D5 input/output (BSC) (initial value)  On-chip ROM enabled extension mode  0: PC5 input/output (port)  1*2: D5 input/output (BSC) (initial value)  Single-chip mode  0: PC5 input/output (port) (initial value)  1: PC5 input/output (port) |  |  |  |  |
| 3 to 1 | _        | All 0            | R   | Reserved These bits are always read as 0. The write value should always be 0.   |  |  |  |  |

| Bit | Bit Name | Initial<br>Value | R/W | Description                                |
|-----|----------|------------------|-----|--|
| 0   | PC4MD    | 0/1*1            | R/W | PC4 Mode                                   |
|     |          |                  |     | Controls the function of the PC4/D4 pin.   |
|     |          |                  |     | On-chip ROM disabled extension mode        |
|     |          |                  |     | 0*2: D4 input/output (BSC)                 |
|     |          |                  |     | 1*2: D4 input/output (BSC) (initial value) |
|     |          |                  |     | On-chip ROM enabled extension mode         |
|     |          |                  |     | 0: PC4 input/output (port)                 |
|     |          |                  |     | 1*2: D4 input/output (BSC) (initial value) |
|     |          |                  |     | Single-chip mode                           |
|     |          |                  |     | 0: PC4 input/output (port) (initial value) |
|     |          |                  |     | 1: PC4 input/output (port)                 |

Notes: 1. The initial value depends on the operating mode of the LSI.

2. The pin is pulled up.

# (4) Port C Control Register 1 (PCCR1)

| Bit:           | 15 | 14 | 13 | 12        | 11 | 10 | 9 | 8         | 7 | 6 | 5 | 4         | 3 | 2 | 1 | 0         |
|----------------|----|----|----|-----------|----|----|---|-----------|---|---|---|-----------|---|---|---|-----------|
|                | -  | -  | -  | PC3<br>MD | -  | -  | - | PC2<br>MD | - | - | - | PC1<br>MD | - | - | - | PC0<br>MD |
| Initial value: | 0  | 0  | 0  | 0/1*1     | 0  | 0  | 0 | 0/1*1     | 0 | 0 | 0 | 0/1*1     | 0 | 0 | 0 | 0/1*1     |
| R/W:           | R  | R  | R  | R/W       | R  | R  | R | R/W       | R | R | R | R/W       | R | R | R | R/W       |

Note: 1. The initial value depends on the operating mode of the LSI.

| Bit      | Bit Name | Initial<br>Value | R/W | Description   |
|----------|----------|------------------|-----|---|
| 15 to 13 | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 12       | PC3MD    | 0/1*1            | R/W | PC3 Mode  |
|          |          |                  |     | Controls the function of the PC3/D3 pin.  |
|          |          |                  |     | <ul> <li>On-chip ROM disabled extension mode         0*²: D3 input/output (BSC)         1*²: D3 input/output (BSC) (initial value)</li> <li>On-chip ROM enabled extension mode         0: PC3 input/output (port)         1*²: D3 input/output (BSC) (initial value)</li> <li>Single-chip mode         0: PC3 input/output (port) (initial value)         1: PC3 input/output (port)</li> </ul> |
| 11 to 9  | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.  |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 8      | PC2MD    | 0/1*1            | R/W | PC2 Mode  Controls the function of the PC2/D2 pin.  On-chip ROM disabled extension mode  0*2: D2 input/output (BSC)  1*2: D2 input/output (BSC) (initial value)  On-chip ROM enabled extension mode  0: PC2 input/output (port)  1*2: D2 input/output (BSC) (initial value)  Single-chip mode  0: PC2 input/output (port) (initial value)  1: PC2 input/output (port) |
| 7 to 5 | _        | All 0            | R   | Reserved These bits are always read as 0. The write value should always be 0.   |
| 4      | PC1MD    | 0/1*1            | R/W | PC1 Mode  Controls the function of the PC1/D1 pin.  On-chip ROM disabled extension mode  0*2: D1 input/output (BSC)  1*2: D1 input/output (BSC) (initial value)  On-chip ROM enabled extension mode  0: PC1 input/output (port)  1*2: D1 input/output (BSC) (initial value)  Single-chip mode  0: PC1 input/output (port) (initial value)  1: PC1 input/output (port) |
| 3 to 1 | _        | All 0            | R   | Reserved These bits are always read as 0. The write value should always be 0.   |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 0   | PC0MD    | 0/1*1            | R/W | PC0 Mode  |
|     |          |                  |     | Controls the function of the PC0/D0 pin.  |
|     |          |                  |     | <ul> <li>On-chip ROM disabled extension mode         0*²: D0 input/output (BSC)         1*²: D0 input/output (BSC) (initial value)</li> <li>On-chip ROM enabled extension mode         0: PC0 input/output (port)         1*²: D0 input/output (BSC) (initial value)</li> </ul> |
|     |          |                  |     | Single-chip mode  |
|     |          |                  |     | 0: PC0 input/output (port) (initial value)  |
|     |          |                  |     | 1: PC0 input/output (port)  |

Notes: 1. The initial value depends on the operating mode of the LSI.

2. The pin is pulled up.

#### 22.1.7 Port D I/O Register (PDIOR)

PDIOR is a 16-bit readable/writable register that sets the I/O direction of the port D pins.

PDIOR is enabled only when the port D pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PDIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13          | 12          | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|----|----|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | -  | -  | PD13<br>IOR | PD12<br>IOR | PD11<br>IOR | PD10<br>IOR | PD9<br>IOR | PD8<br>IOR | PD7<br>IOR | PD6<br>IOR | PD5<br>IOR | PD4<br>IOR | PD3<br>IOR | PD2<br>IOR | PD1<br>IOR | PD0<br>IOR |
| Initial value: | 0  | 0  | 0           | 0           | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R  | R  | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

| Bit     | Bit Name             | Initial<br>Value | R/W | Description  |
|---------|----------------------|------------------|-----|--|
| 15, 14  | _                    | All 0            | R   | Reserved   |
|         |                      |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 13 to 0 | PD13IOR to<br>PD0IOR | All 0            | R/W | The PD13IOR to PD0IOR bits correspond to the PD13 to PC0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|         |                      |                  |     | 0: The corresponding pin is set to input.  |
|         |                      |                  |     | 1: The corresponding pin is set to output.   |

## 22.1.8 Port D Control Registers 1 and 2 (PDCR1 and PDCR2)

PDCR1 and PDCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port D.

PDCR1 and PDCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

# (1) Port D Control Register 2 (PDCR2)

| Bit:           | 15 | 14 | 13 | 12 | 11    | 10      | 9     | 8        | 7     | 6       | 5     | 4        | 3    | 2       | 1    | 0       |
|----------------|----|----|----|----|-------|---------|-------|----------|-------|---------|-------|----------|------|---------|------|---------|
|                | -  | -  | -  | -  | PD13N | ИD[1:0] | PD12N | /ID[1:0] | PD11N | ИD[1:0] | PD10N | /ID[1:0] | PD9M | ID[1:0] | PD8N | ID[1:0] |
| Initial value: | 0  | 0  | 0  | 0  | 0     | 0       | 0     | 0        | 0     | 0       | 0     | 0        | 0    | 0       | 0    | 0       |
| R/W:           | R  | R  | R  | R  | R/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<br>Value | R/W                          | Description  |
|----------|----------|------------------|------------------------------|--|
| 15 to 12 | _        | All 0            | R                            | Reserved   |
|          |          |                  |                              | These bits are always read as 0. The write value should always be 0. |
| 11, 10   | PD13MD   | 00               | R/W                          | PD13 Mode  |
|          | [1:0]    |                  |                              | Control the function of the PD13/TCLKB/TIJ1 pin.                     |
|          |          |                  | 00: PD13 input/output (port) |  |
|          |          |                  |                              | 01: TCLKB input (ATU-III)  |
|          |          |                  |                              | 10: Setting prohibited   |
|          |          |                  |                              | 11: TIJ1 input (ATU-III)   |
| 9, 8     | PD12MD   | 00               | R/W                          | PD12 Mode  |
|          | [1:0]    |                  |                              | Control the function of the PD12/TCLKA/TIOC41/TIJ0 pin.              |
|          |          |                  |                              | 00: PD12 input/output (port)   |
|          |          |                  |                              | 01: TCLKA input (ATU-III)  |
|          |          |                  |                              | 10: TIOC41 input/output (ATU-III)                                    |
|          |          |                  |                              | 11: TIJ0 input (ATU-III)   |

| 7, 6 PD11MD 00 R/W PD11 Mode [1:0] Control the function of the PD11/TIOC23/TIF2B/TOE51 pin. |  |
|---|--|
| Control the function of the   |  |
|   |  |
| 00: PD11 input/output (port)  |  |
| 01: TIOC23 input/output (ATU-III)   |  |
| 10: TIF2B input (ATU-III)   |  |
| 11: TOE51 output (ATU-III)  |  |
| 5, 4 PD10MD 00 R/W PD10 Mode  |  |
| [1:0] Control the function of the PD10/TIOC22/TIF1B/TOE50 pin.                              |  |
| 00: PD10 input/output (port)  |  |
| 01: TIOC22 input/output (ATU-III)   |  |
| 10: TIF1B input (ATU-III)   |  |
| 11: TOE50 output (ATU-III)  |  |
| 3, 2 PD9MD 00 R/W PD9 Mode  |  |
| [1:0] Control the function of the PD9/TIOC21/TIF0B/TOE43 pin.                               |  |
| 00: PD9 input/output (port)   |  |
| 01: TIOC21 input/output (ATU-III)   |  |
| 10: TIF0B input (ATU-III)   |  |
| 11: TOE43 output (ATU-III)  |  |
| 1, 0 PD8MD 00 R/W PD8 Mode  |  |
| [1:0] Control the function of the PD8/TIOC20/TIOC33/TOE53 pin.                              |  |
| 00: PD8 input/output (port)   |  |
| 01: TIOC20 input/output (ATU-III)   |  |
| 10: TIOC33 input/output (ATU-III)   |  |
| 11: TOE53 output (ATU-III)  |  |

# (2) Port D Control Register 1 (PDCR1)

| Bit:           | 15   | 14     | 13   | 12     | 11   | 10     | 9    | 8       | 7    | 6       | 5    | 4       | 3    | 2      | 1    | 0       |
|----------------|------|--------|------|--------|------|--------|------|---------|------|---------|------|---------|------|--------|------|---------|
|                | PD7M | D[1:0] | PD6M | D[1:0] | PD5M | D[1:0] | PD4M | ID[1:0] | PD3N | 1D[1:0] | PD2N | ID[1:0] | PD1M | D[1:0] | PD0N | ID[1:0] |
| Initial value: | 0    | 0      | 0    | 0      | 0    | 0      | 0    | 0       | 0    | 0       | 0    | 0       | 0    | 0      | 0    | 0       |
| R/W:           | R/W  | R/W    | R/W  | R/W    | R/W  | R/W    | R/W  | R/W     | R/W  | R/W     | R/W  | R/W     | R/W  | R/W    | R/W  | R/W     |

|        |          | Initial |     |  |
|--------|----------|---------|-----|--|
| Bit    | Bit Name | Value   | R/W | Description  |
| 15, 14 | PD7MD    | 00      | R/W | PD7 Mode   |
|        | [1:0]    |         |     | Control the function of the PD7/TIOC13/TOE42 pin.        |
|        |          |         |     | 00: PD7 input/output (port)                              |
|        |          |         |     | 01: TIOC13 input/output (ATU-III)                        |
|        |          |         |     | 10: Setting prohibited                                   |
|        |          |         |     | 11: TOE42 output (ATU-III)                               |
| 13, 12 | PD6MD    | 00      | R/W | PD6 Mode   |
|        | [1:0]    |         |     | Control the function of the PD6/TIOC12/TOE41 pin.        |
|        |          |         |     | 00: PD6 input/output (port)                              |
|        |          |         |     | 01: TIOC12 input/output (ATU-III)                        |
|        |          |         |     | 10: Setting prohibited                                   |
|        |          |         |     | 11: TOE41 output (ATU-III)                               |
| 11, 10 | PD5MD    | 00      | R/W | PD5 Mode   |
|        | [1:0]    |         |     | Control the function of the PD5/TIOC11/TOE23/TOE40 pin.  |
|        |          |         |     | 00: PD5 input/output (port)                              |
|        |          |         |     | 01: TIOC11 input/output (ATU-III)                        |
|        |          |         |     | 10: TOE23 output (ATU-III)                               |
|        |          |         |     | 11: TOE40 output (ATU-III)                               |
| 9, 8   | PD4MD    | 00      | R/W | PD4 Mode   |
|        | [1:0]    |         |     | Control the function of the PD4/TIOC10/TIOC32/TOE52 pin. |
|        |          |         |     | 00: PD4 input/output (port)                              |
|        |          |         |     | 01: TIOC10 input/output (ATU-III)                        |
|        |          |         |     | 10: TIOC32 input/output (ATU-III)                        |
|        |          |         |     | 11: TOE52 output (ATU-III)                               |
|        |          |         |     |  |

| Bit  | Bit Name | Initial<br>Value | R/W                         | Description   |
|------|----------|------------------|-----------------------------|---|
| 7, 6 | PD3MD    | 00               | R/W                         | PD3 Mode  |
|      | [1:0]    |                  |                             | Control the function of the PD3/TIOC03/TOE22/TOE53 pin. |
|      |          |                  |                             | 00: PD3 input/output (port)                             |
|      |          |                  |                             | 01: TIOC03 input/output (ATU-III)                       |
|      |          |                  |                             | 10: TOE22 output (ATU-III)                              |
|      |          |                  |                             | 11: TOE53 output (ATU-III)                              |
| 5, 4 | PD2MD    | 00               | R/W                         | PD2 Mode  |
|      | [1:0]    |                  |                             | Control the function of the PD2/TIOC02/TOE21/TOE52 pin. |
|      |          |                  | 00: PD2 input/output (port) |   |
|      |          |                  |                             | 01: TIOC02 input/output (ATU-III)                       |
|      |          |                  |                             | 10: TOE21 output (ATU-III)                              |
|      |          |                  |                             | 11: TOE52 output (ATU-III)                              |
| 3, 2 | PD1MD    | 00               | R/W                         | PD1 Mode  |
|      | [1:0]    |                  |                             | Control the function of the PD1/TIOC01/TOE20 pin.       |
|      |          |                  |                             | 00: PD1 input/output (port)                             |
|      |          |                  |                             | 01: TIOC01 input/output (ATU-III)                       |
|      |          |                  |                             | 10: TOE20 output (ATU-III)                              |
|      |          |                  |                             | 11: Setting prohibited                                  |
| 1, 0 | PD0MD    | 00               | R/W                         | PD0 Mode  |
|      | [1:0]    |                  |                             | Control the function of the PD0/TIOC00/TIOC31 pin.      |
|      |          |                  |                             | 00: PD0 input/output (port)                             |
|      |          |                  |                             | 01: TIOC00 input/output (ATU-III)                       |
|      |          |                  |                             | 10: TIOC31 input/output (ATU-III)                       |
|      |          |                  |                             | 11: Setting prohibited                                  |

### 22.1.9 Port E I/O Register (PEIOR)

PEIOR is a 16-bit readable/writable register that sets the I/O direction of the port E pins.

PEIOR is enabled only when the port E pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PEIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13          | 12          | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|----|----|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | -  | -  | PE13<br>IOR | PE12<br>IOR | PE11<br>IOR | PE10<br>IOR | PE9<br>IOR | PE8<br>IOR | PE7<br>IOR | PE6<br>IOR | PE5<br>IOR | PE4<br>IOR | PE3<br>IOR | PE2<br>IOR | PE1<br>IOR | PE0<br>IOR |
| Initial value: | 0  | 0  | 0           | 0           | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R  | R  | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

| Bit     | Bit Name             | Initial<br>Value | R/W | Description  |
|---------|----------------------|------------------|-----|--|
| 15, 14  | _                    | All 0            | R   | Reserved   |
|         |                      |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 13 to 0 | PE13IOR to<br>PE0IOR | All 0            | R/W | The PE13IOR to PE0IOR bits correspond to the PE13 to PE0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|         |                      |                  |     | 0: The corresponding pin is set to input.  |
|         |                      |                  |     | 1: The corresponding pin is set to output.   |

# Port E Control Registers 1 and 2 (PECR1 and PECR2)

PECR1 and PECR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port E.

PECR1 and PECR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

#### **Port E Control Register 2 (PECR2) (1)**

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10         | 9 | 8          | 7 | 6          | 5 | 4          | 3 | 2         | 1 | 0         |
|----------------|----|----|----|----|----|------------|---|------------|---|------------|---|------------|---|-----------|---|-----------|
|                | -  | -  | -  | -  | -  | PE13<br>MD | - | PE12<br>MD | - | PE11<br>MD | - | PE10<br>MD | - | PE9<br>MD | - | PE8<br>MD |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0          | 0 | 0          | 0 | 0          | 0 | 0          | 0 | 0         | 0 | 0         |
| R/W:           | R  | R  | R  | R  | R  | R/W        | R | R/W        | R | R/W        | R | R/W        | R | R/W       | R | R/W       |

| Bit      | Bit Name | Initial<br>Value | R/W | Description  |
|----------|----------|------------------|-----|--|
| 15 to 11 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 10       | PE13MD   | 0                | R/W | PE13 Mode  |
|          |          |                  |     | Controls the function of the PE13/TOE13 pin.                         |
|          |          |                  |     | 0: PE13 input/output (port)  |
|          |          |                  |     | 1: TOE13 output (ATU-III)  |
| 9        | _        | 0                | R   | Reserved   |
|          |          |                  |     | This bit is always read as 0. The write value should always be 0.    |
| 8        | PE12MD   | 0                | R/W | PE12 Mode  |
|          |          |                  |     | Controls the function of the PE12/TOE12 pin.                         |
|          |          |                  |     | 0: PE12 input/output (port)  |
|          |          |                  |     | 1: TOE12 output (ATU-III)  |
| 7        | _        | 0                | R   | Reserved   |
|          |          |                  |     | This bit is always read as 0. The write value should always be 0.    |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 6   | PE11MD   | 0                | R/W | PE11 Mode   |
|     |          |                  |     | Controls the function of the PE11/TOE11 pin.                      |
|     |          |                  |     | 0: PE11 input/output (port)                                       |
|     |          |                  |     | 1: TOE11 output (ATU-III)   |
| 5   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 4   | PE10MD   | 0                | R/W | PE10 Mode   |
|     |          |                  |     | Controls the function of the PE10/TOE10 pin.                      |
|     |          |                  |     | 0: PE10 input/output (port)                                       |
|     |          |                  |     | 1: TOE10 output (ATU-III)   |
| 3   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 2   | PE9MD    | 0                | R/W | PE9 Mode  |
|     |          |                  |     | Controls the function of the PE9/TOE03 pin.                       |
|     |          |                  |     | 0: PE9 input/output (port)  |
|     |          |                  |     | 1: TOE03 output (ATU-III)   |
| 1   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 0   | PE8MD    | 0                | R/W | PE8 Mode  |
|     |          |                  |     | Controls the function of the PE8/TOE02 pin.                       |
|     |          |                  |     | 0: PE8 input/output (port)  |
|     |          |                  |     | 1: TOE02 output (ATU-III)   |

# (2) Port E Control Register 1 (PECR1)

| Bit:           | 15   | 14     | 13   | 12     | 11 | 10        | 9 | 8         | 7 | 6         | 5    | 4       | 3    | 2       | 1 | 0         |
|----------------|------|--------|------|--------|----|-----------|---|-----------|---|-----------|------|---------|------|---------|---|-----------|
|                | PE7M | D[1:0] | PE6M | D[1:0] | -  | PE5<br>MD | - | PE4<br>MD | - | PE3<br>MD | PE2N | ID[1:0] | PE1M | ID[1:0] | - | PE0<br>MD |
| Initial value: | 0    | 0      | 0    | 0      | 0  | 0         | 0 | 0         | 0 | 0         | 0    | 0       | 0    | 0       | 0 | 0         |
| R/W:           | R/W  | R/W    | R/W  | R/W    | R  | R/W       | R | R/W       | R | R/W       | R/W  | R/W     | R/W  | R/W     | R | R/W       |

| Bit    | Bit Name       | Initial<br>Value | R/W | Description   |
|--------|----------------|------------------|-----|---|
|        |                |                  |     | Description   |
| 15, 14 | PE7MD<br>[1:0] | 00               | R/W | PE7 Mode  |
|        | [1.0]          |                  |     | Control the function of the PE7/TOE01/CRx_B pin.                  |
|        |                |                  |     | 00: PE7 input/output (port)                                       |
|        |                |                  |     | 01: TOE01 output (ATU-III)  |
|        |                |                  |     | 10: CRx_B input (RCAN-TL1)  |
|        |                |                  |     | 11: Setting prohibited  |
| 13, 12 | PE6MD          | 00               | R/W | PE6 Mode  |
|        | [1:0]          |                  |     | Control the function of the PE6/TOE00/CTx_B pin.                  |
|        |                |                  |     | 00: PE6 input/output (port)                                       |
|        |                |                  |     | 01: TOE00 output (ATU-III)  |
|        |                |                  |     | 10: CTx_B output (RCAN-TL1)                                       |
|        |                |                  |     | 11: Setting prohibited  |
| 11     | _              | 0                | R   | Reserved  |
|        |                |                  |     | This bit is always read as 0. The write value should always be 0. |
| 10     | PE5MD          | 0                | R/W | PE5 Mode  |
|        |                |                  |     | Controls the function of the PE5/TIA05 pin.                       |
|        |                |                  |     | 0: PE5 input/output (port)  |
|        |                |                  |     | 1: TIA05 input (ATU-III)  |
| 9      | _              | 0                | R   | Reserved  |
|        |                |                  |     | This bit is always read as 0. The write value should always be 0. |
| 8      | PE4MD          | 0                | R/W | PE4 Mode  |
|        |                |                  |     | Controls the function of the PE4/TIA04 pin.                       |
|        |                |                  |     | 0: PE4 input/output (port)  |
|        |                |                  |     | 1: TIA04 input (ATU-III)  |
|        |                |                  |     |   |

| Bit Bit Name Value R/W Description  7 — 0 R Reserved This bit is always read as 0. The write value always be 0.  6 PE3MD 0 R/W PE3 Mode |             |
|---|-------------|
| always be 0.  |             |
| 6 PE3MD 0 R/W PE3 Mode  | oin.        |
|   | oin.        |
| Controls the function of the PE3/TIA03 p  |             |
| 0: PE3 input/output (port)  |             |
| 1: TIA03 input (ATU-III)  |             |
| 5, 4 PE2MD 00 R/W PE2 Mode  |             |
| [1:0] Control the function of the PE2/TIA02/TIOC43/TIOC30 pin.  |             |
| 00: PE2 input/output (port)   |             |
| 01: TIA02 input (ATU-III)   |             |
| 10: TIOC43 input/output (ATU-III)   |             |
| 11: TIOC30 input/output (ATU-III)   |             |
| 3, 2 PE1MD 00 R/W PE1 Mode  |             |
| [1:0] Control the function of the PE1/TIA01/TIOC42/TIOC40 pin.  |             |
| 00: PE1 input/output (port)   |             |
| 01: TIA01 input (ATU-III)   |             |
| 10: TIOC42 input/output (ATU-III)   |             |
| 11: TIOC40 input/output (ATU-III)   |             |
| 1 — 0 R Reserved  |             |
| This bit is always read as 0. The write va always be 0.   | alue should |
| 0 PE0MD 0 R/W PE0 Mode  |             |
| Controls the function of the PE0/TIA00 p  | oin.        |
| 0: PE0 input/output (port)  |             |
| 1: TIA00 input (ATU-III)  |             |

#### Port F I/O Register (PFIOR) 22.1.11

PFIOR is a 16-bit readable/writable register that sets the I/O direction of the port F pins.

PFIOR is enabled only when the port F pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PFIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | PF15<br>IOR | PF14<br>IOR | PF13<br>IOR | PF12<br>IOR | PF11<br>IOR | PF10<br>IOR | PF9<br>IOR | PF8<br>IOR | PF7<br>IOR | PF6<br>IOR | PF5<br>IOR | PF4<br>IOR | PF3<br>IOR | PF2<br>IOR | PF1<br>IOR | PF0<br>IOR |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

|         |                      | Initial |     |  |
|---------|----------------------|---------|-----|--|
| Bit     | Bit Name             | Value   | R/W | Description  |
| 15 to 0 | PF15IOR to<br>PF0IOR | All 0   | R/W | The PF15IOR to PF0IOR bits correspond to the PF15 to PF0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|         |                      |         |     | 0: The corresponding pin is set to input.  |
|         |                      |         |     | 1: The corresponding pin is set to output.   |

## 22.1.12 Port F Control Registers 1 and 2 (PFCR1 and PFCR2)

PFCR1 and PFCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port F.

PFCR1 and PFCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

### (1) Port F Control Register 2 (PFCR2)

Initial

| Bit:           | 15    | 14       | 13    | 12      | 11    | 10       | 9     | 8       | 7     | 6        | 5     | 4       | 3    | 2      | 1    | 0      |
|----------------|-------|----------|-------|---------|-------|----------|-------|---------|-------|----------|-------|---------|------|--------|------|--------|
|                | PF15N | /ID[1:0] | PF14N | 1D[1:0] | PF13N | /ID[1:0] | PF12N | 1D[1:0] | PF11N | /ID[1:0] | PF10N | 1D[1:0] | PF9M | D[1:0] | PF8M | D[1:0] |
| Initial value: | 0     | 0        | 0     | 0       | 0     | 0        | 0     | 0       | 0     | 0        | 0     | 0       | 0    | 0      | 0    | 0      |
| R/W·           | R/W   | R/W      | R/W   | R/W     | R/W   | R/W      | R/W   | R/W     | R/W   | R/W      | R/W   | R/W     | R/W  | R/W    | R/W  | R/W    |

| Bit    | Bit Name | Value | R/W | Description  |
|--------|----------|-------|-----|--|
| 15, 14 | PF15MD   | 00    | R/W | PF15 Mode  |
|        | [1:0]    |       |     | Control the function of the PF15/TOD33B/CRx_B/RxD_A pin. |
|        |          |       |     | 00: PF15 input/output (port)                             |
|        |          |       |     | 01: TOD33B output (ATU-III)                              |
|        |          |       |     | 10: CRx_B input (RCAN-TL1)                               |
|        |          |       |     | 11: RxD_A input (SCI)                                    |
| 13, 12 | PF14MD   | 00    | R/W | PF14 Mode  |
|        | [1:0]    |       |     | Control the function of the PF14/TOD32B/CTx_B/TxD_A pin. |
|        |          |       |     | 00: PF14 input/output (port)                             |
|        |          |       |     | 01: TOD32B output (ATU-III)                              |
|        |          |       |     | 10: CTx_B output (RCAN-TL1)                              |
|        |          |       |     | 11: TxD_A output (SCI)                                   |
| 11, 10 | PF13MD   | 00    | R/W | PF13 Mode  |
|        | [1:0]    |       |     | Control the function of the PF13/TOD31B/TIF19 pin.       |
|        |          |       |     | 00: PF13 input/output (port)                             |
|        |          |       |     | 01: TOD31B output (ATU-III)                              |
|        |          |       |     | 10: Setting prohibited                                   |
|        |          |       |     | 11: TIF19 input (ATU-III)                                |

| Bit  | Bit Name        | Initial<br>Value | R/W | Description  |  |  |  |  |
|------|-----------------|------------------|-----|--|--|--|--|--|
| 9, 8 | PF12MD          | 00               | R/W | PF12 Mode  |  |  |  |  |
|      | [1:0]           |                  |     | Control the function of the PF12/TOD30B/TIF18 pin. |  |  |  |  |
|      |                 |                  |     | 00: PF12 input/output (port)                       |  |  |  |  |
|      |                 |                  |     | 01: TOD30B output (ATU-III)                        |  |  |  |  |
|      |                 |                  |     | 10: Setting prohibited                             |  |  |  |  |
|      |                 |                  |     | 11: TIF18 input (ATU-III)                          |  |  |  |  |
| 7, 6 | PF11MD          | 00               | R/W | PF11 Mode  |  |  |  |  |
|      | [1:0]           |                  |     | Control the function of the PF11/TOD23B/TIF17 pin  |  |  |  |  |
|      |                 |                  |     | 00: PF11 input/output (port)                       |  |  |  |  |
|      |                 |                  |     | 01: TOD23B output (ATU-III)                        |  |  |  |  |
|      |                 |                  |     | 10: Setting prohibited                             |  |  |  |  |
|      |                 |                  |     | 11: TIF17 input (ATU-III)                          |  |  |  |  |
| 5, 4 | PF10MD<br>[1:0] | 00               | R/W | PF10 Mode  |  |  |  |  |
|      |                 |                  |     | Control the function of the PF10/TOD22B/TIF16 pin. |  |  |  |  |
|      |                 |                  |     | 00: PF10 input/output (port)                       |  |  |  |  |
|      |                 |                  |     | 01: TOD22B output (ATU-III)                        |  |  |  |  |
|      |                 |                  |     | 10: Setting prohibited                             |  |  |  |  |
|      |                 |                  |     | 11: TIF16 input (ATU-III)                          |  |  |  |  |
| 3, 2 | PF9MD           | 00               | R/W | PF9 Mode   |  |  |  |  |
|      | [1:0]           |                  |     | Control the function of the PF9/TOD21B/TIF15 pin.  |  |  |  |  |
|      |                 |                  |     | 00: PF9 input/output (port)                        |  |  |  |  |
|      |                 |                  |     | 01: TOD21B output (ATU-III)                        |  |  |  |  |
|      |                 |                  |     | 10: Setting prohibited                             |  |  |  |  |
|      |                 |                  |     | 11: TIF15 input (ATU-III)                          |  |  |  |  |
| 1, 0 | PF8MD           | 00               | R/W | PF8 Mode   |  |  |  |  |
|      | [1:0]           |                  |     | Control the function of the PF8/TOD20B/TIF14 pin.  |  |  |  |  |
|      |                 |                  |     | 00: PF8 input/output (port)                        |  |  |  |  |
|      |                 |                  |     | 01: TOD20B output (ATU-III)                        |  |  |  |  |
|      |                 |                  |     | 10: Setting prohibited                             |  |  |  |  |
|      |                 |                  |     | 11: TIF14 input (ATU-III)                          |  |  |  |  |

# (2) Port F Control Register 1 (PFCR1)

| Bit:           | 15   | 14     | 13   | 12     | 11   | 10     | 9    | 8      | 7    | 6       | 5    | 4      | 3    | 2      | 1    | 0      |
|----------------|------|--------|------|--------|------|--------|------|--------|------|---------|------|--------|------|--------|------|--------|
|                | PF7M | D[1:0] | PF6M | D[1:0] | PF5M | D[1:0] | PF4M | D[1:0] | PF3M | ID[1:0] | PF2M | D[1:0] | PF1M | D[1:0] | PF0M | D[1:0] |
| Initial value: | 0    | 0      | 0    | 0      | 0    | 0      | 0    | 0      | 0    | 0       | 0    | 0      | 0    | 0      | 0    | 0      |
| R/W:           | R/W  | R/W     | R/W  | R/W    | R/W  | R/W    | R/W  | R/W    |

| Bit    | Bit Name       | Initial<br>Value | R/W | Description                                       |  |  |  |  |
|--------|----------------|------------------|-----|---|--|--|--|--|
| 15, 14 | PF7MD          | 00               | R/W | PF7 Mode  |  |  |  |  |
|        | [1:0]          |                  |     | Control the function of the PF7/TOD13B/TIF13 pin. |  |  |  |  |
|        |                |                  |     | 00: PF7 input/output (port)                       |  |  |  |  |
|        |                |                  |     | 01: TOD13B output (ATU-III)                       |  |  |  |  |
|        |                |                  |     | 10: Setting prohibited                            |  |  |  |  |
|        |                |                  |     | 11: TIF13 input (ATU-III)                         |  |  |  |  |
| 13, 12 | PF6MD<br>[1:0] | 00               | R/W | PF6 Mode  |  |  |  |  |
|        |                |                  |     | Control the function of the PF6/TOD12B/TIF12 pin. |  |  |  |  |
|        |                |                  |     | 00: PF6 input/output (port)                       |  |  |  |  |
|        |                |                  |     | 01: TOD12B output (ATU-III)                       |  |  |  |  |
|        |                |                  |     | 10: Setting prohibited                            |  |  |  |  |
|        |                |                  |     | 11: TIF12 input (ATU-III)                         |  |  |  |  |
| 11, 10 | PF5MD          | 00               | R/W | PF5 Mode  |  |  |  |  |
|        | [1:0]          |                  |     | Control the function of the PF5/TOD11B/TIF11 pin. |  |  |  |  |
|        |                |                  |     | 00: PF5 input/output (port)                       |  |  |  |  |
|        |                |                  |     | 01: TOD11B output (ATU-III)                       |  |  |  |  |
|        |                |                  |     | 10: Setting prohibited                            |  |  |  |  |
|        |                |                  |     | 11: TIF11 input (ATU-III)                         |  |  |  |  |
| 9, 8   | PF4MD          | 00               | R/W | PF4 Mode  |  |  |  |  |
|        | [1:0]          |                  |     | Control the function of the PF4/TOD10B/TIF10 pin. |  |  |  |  |
|        |                |                  |     | 00: PF4 input/output (port)                       |  |  |  |  |
|        |                |                  |     | 01: TOD10B output (ATU-III)                       |  |  |  |  |
|        |                |                  |     | 10: Setting prohibited                            |  |  |  |  |
|        |                |                  |     | 11: TIF10 input (ATU-III)                         |  |  |  |  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description                                      |  |  |  |  |  |
|------|----------|------------------|-----|--|--|--|--|--|--|
| 7, 6 | PF3MD    | 00               | R/W | PF3 Mode   |  |  |  |  |  |
|      | [1:0]    |                  |     | Control the function of the PF3/TOD03B/TIF9 pin. |  |  |  |  |  |
|      |          |                  |     | 00: PF3 input/output (port)                      |  |  |  |  |  |
|      |          |                  |     | 01: TOD03B output (ATU-III)                      |  |  |  |  |  |
|      |          |                  |     | 10: Setting prohibited                           |  |  |  |  |  |
|      |          |                  |     | 11: TIF9 input (ATU-III)                         |  |  |  |  |  |
| 5, 4 | PF2MD    | 00               | R/W | PF2 Mode   |  |  |  |  |  |
|      | [1:0]    |                  |     | Control the function of the PF2/TOD02B/TIF8 pin. |  |  |  |  |  |
|      |          |                  |     | 00: PF2 input/output (port)                      |  |  |  |  |  |
|      |          |                  |     | 01: TOD02B output (ATU-III)                      |  |  |  |  |  |
|      |          |                  |     | 10: Setting prohibited                           |  |  |  |  |  |
|      |          |                  |     | 11: TIF8 input (ATU-III)                         |  |  |  |  |  |
| 3, 2 | PF1MD    | 00               | R/W | PF1 Mode   |  |  |  |  |  |
|      | [1:0]    |                  |     | Control the function of the PF1/TOD01B/TIF7 pin. |  |  |  |  |  |
|      |          |                  |     | 00: PF1 input/output (port)                      |  |  |  |  |  |
|      |          |                  |     | 01: TOD01B output (ATU-III)                      |  |  |  |  |  |
|      |          |                  |     | 10: Setting prohibited                           |  |  |  |  |  |
|      |          |                  |     | 11: TIF7 input (ATU-III)                         |  |  |  |  |  |
| 1, 0 | PF0MD    | 00               | R/W | PF0 Mode   |  |  |  |  |  |
|      | [1:0]    |                  |     | Control the function of the PF0/TOD00B/TIF6 pin. |  |  |  |  |  |
|      |          |                  |     | 00: PF0 input/output (port)                      |  |  |  |  |  |
|      |          |                  |     | 01: TOD00B output (ATU-III)                      |  |  |  |  |  |
|      |          |                  |     | 10: Setting prohibited                           |  |  |  |  |  |
|      |          |                  |     | 11: TIF6 input (ATU-III)                         |  |  |  |  |  |

Page 1122 of 1824

# 22.1.13 Port G I/O Register (PGIOR)

PGIOR is a 16-bit readable/writable register that sets the I/O direction of the port G pins.

PGIOR is enabled only when the port G pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PGIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | PG15<br>IOR | PG14<br>IOR | PG13<br>IOR | PG12<br>IOR | PG11<br>IOR | PG10<br>IOR | PG9<br>IOR | PG8<br>IOR | PG7<br>IOR | PG6<br>IOR | PG5<br>IOR | PG4<br>IOR | PG3<br>IOR | PG2<br>IOR | PG1<br>IOR | PG0<br>IOR |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

|         |                      | Initial |     |  |
|---------|----------------------|---------|-----|--|
| Bit     | Bit Name             | Value   | R/W | Description  |
| 15 to 0 | PG15IOR to<br>PG0IOR | All 0   | R/W | The PG15IOR to PG0IOR bits correspond to the PG15 to PG0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|         |                      |         |     | 0: The corresponding pin is set to input.  |
|         |                      |         |     | 1: The corresponding pin is set to output.   |

### Port G Control Registers 1 and 2 (PGCR1 and PGCR2)

PGCR1 and PGCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port G.

PGCR1 and PGCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

#### Port G Control Register 2 (PGCR2) **(1)**

| Bit:           | 15    | 14       | 13    | 12      | 11    | 10       | 9     | 8        | 7     | 6       | 5     | 4        | 3    | 2       | 1    | 0       |
|----------------|-------|----------|-------|---------|-------|----------|-------|----------|-------|---------|-------|----------|------|---------|------|---------|
|                | PG15N | /ID[1:0] | PG14N | 1D[1:0] | PG13N | /ID[1:0] | PG12N | /ID[1:0] | PG11N | ИD[1:0] | PG10N | /ID[1:0] | PG9M | ID[1:0] | PG8M | ID[1:0] |
| Initial value: | 0     | 0        | 0     | 0       | 0     | 0        | 0     | 0        | 0     | 0       | 0     | 0        | 0    | 0       | 0    | 0       |
| R/W:           | R/W   | R/W      | R/W   | R/W     | R/W   | R/W      | R/W   | R/W      | R/W   | R/W     | R/W   | R/W      | R/W  | R/W     | R/W  | R/W     |

|          | Initial         |  |  |
|----------|-----------------|--|--|
| Bit Name | Value           | R/W                                      | Description  |
| PG15MD   | 00              | R/W                                      | PG15 Mode  |
| [1:0]    |                 |  | Control the function of the PG15/TOD33A/SSLA7/TIF13 pin.   |
|          |                 |  | 00: PG15 input/output (port)   |
|          |                 |  | 01: TOD33A output (ATU-III)  |
|          |                 |  | 10: SSLA7 output (RSPI)  |
|          |                 |  | 11: TIF13 input (ATU-III)  |
| PG14MD   | 00              | R/W                                      | PG14 Mode  |
| [1:0]    |                 |  | Control the function of the PG14/TOD32A/SSLA6/TIF12 pin.   |
|          |                 |  | 00: PG14 input/output (port)   |
|          |                 |  | 01: TOD32A output (ATU-III)  |
|          |                 |  | 10: SSLA6 output (RSPI)  |
|          |                 |  | 11: TIF12 input (ATU-III)  |
|          | PG15MD<br>[1:0] | Bit Name Value PG15MD 00 [1:0] PG14MD 00 | Bit Name         Value         R/W           PG15MD [1:0]         00         R/W           PG14MD         00         R/W |

Apr 01, 2014

|        |          | Initial |     |  |
|--------|----------|---------|-----|--|
| Bit    | Bit Name | Value   | R/W | Description  |
| 11, 10 | PG13MD   | 00      | R/W | PG13 Mode  |
|        | [1:0]    |         |     | Control the function of the PG13/TOD31A/SSLA5/TIF11 pin. |
|        |          |         |     | 00: PG13 input/output (port)                             |
|        |          |         |     | 01: TOD31A output (ATU-III)                              |
|        |          |         |     | 10: SSLA5 output (RSPI)                                  |
|        |          |         |     | 11: TIF11 input (ATU-III)                                |
| 9, 8   | PG12MD   | 00      | R/W | PG12 Mode  |
|        | [1:0]    |         |     | Control the function of the PG12/TOD30A/SSLA4/TIF10 pin. |
|        |          |         |     | 00: PG12 input/output (port)                             |
|        |          |         |     | 01: TOD30A output (ATU-III)                              |
|        |          |         |     | 10: SSLA4 output (RSPI)                                  |
|        |          |         |     | 11: TIF10 input (ATU-III)                                |
| 7, 6   | PG11MD   | 00      | R/W | PG11 Mode  |
|        | [1:0]    |         |     | Control the function of the PG11/TOD23A/SSLC2/TIF9 pin.  |
|        |          |         |     | 00: PG11 input/output (port)                             |
|        |          |         |     | 01: TOD23A output (ATU-III)                              |
|        |          |         |     | 10: SSLC2 output (RSPI)                                  |
|        |          |         |     | 11: TIF9 input (ATU-III)                                 |
| 5, 4   | PG10MD   | 00      | R/W | PG10 Mode  |
|        | [1:0]    |         |     | Control the function of the PG10/TOD22A/SSLC1/TIF8 pin.  |
|        |          |         |     | 00: PG10 input/output (port)                             |
|        |          |         |     | 01: TOD22A output (ATU-III)                              |
|        |          |         |     | 10: SSLC1 output (RSPI)                                  |
|        |          |         |     | 11: TIF8 input (ATU-III)                                 |
|        |          |         |     |  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 3, 2 | PG9MD    | 00               | R/W | PG9 Mode   |
|      | [1:0]    |                  |     | Control the function of the PG9/TOD21A/SSLC0/TIF7 pin. |
|      |          |                  |     | 00: PG9 input/output (port)                            |
|      |          |                  |     | 01: TOD21A output (ATU-III)                            |
|      |          |                  |     | 10: SSLC0 input/output (RSPI)                          |
|      |          |                  |     | 11: TIF7 input (ATU-III)                               |
| 1, 0 | PG8MD    | 00               | R/W | PG8 Mode   |
|      | [1:0]    |                  |     | Control the function of the PG8/TOD20A/SSLB2/TIF6 pin. |
|      |          |                  |     | 00: PG8 input/output (port)                            |
|      |          |                  |     | 01: TOD20A output (ATU-III)                            |
|      |          |                  |     | 10: SSLB2 output (RSPI)                                |
|      |          |                  |     | 11: TIF6 input (ATU-III)                               |

# (2) Port G Control Register 1 (PGCR1)

| Bit:           | 15   | 14     | 13   | 12     | 11   | 10     | 9    | 8      | 7    | 6       | 5    | 4       | 3    | 2       | 1    | 0       |
|----------------|------|--------|------|--------|------|--------|------|--------|------|---------|------|---------|------|---------|------|---------|
|                | PG7M | D[1:0] | PG6M | D[1:0] | PG5M | D[1:0] | PG4M | D[1:0] | PG3M | ID[1:0] | PG2N | ID[1:0] | PG1M | ID[1:0] | PG0N | ID[1:0] |
| Initial value: | 0    | 0      | 0    | 0      | 0    | 0      | 0    | 0      | 0    | 0       | 0    | 0       | 0    | 0       | 0    | 0       |
| R/W:           | R/W  | R/W     | R/W  | R/W     | R/W  | R/W     | R/W  | R/W     |

|        |          | Initial |     |   |
|--------|----------|---------|-----|---|
| Bit    | Bit Name | Value   | R/W | Description   |
| 15, 14 | PG7MD    | 00      | R/W | PG7 Mode  |
|        | [1:0]    |         |     | Control the function of the PG7/TOD13A/SSLB1 pin.       |
|        |          |         |     | 00: PG7 input/output (port)                             |
|        |          |         |     | 01: TOD13A output (ATU-III)                             |
|        |          |         |     | 10: SSLB1 output (RSPI)                                 |
|        |          |         |     | 11: Setting prohibited                                  |
| 13, 12 | PG6MD    | 00      | R/W | PG6 Mode  |
|        | [1:0]    |         |     | Control the function of the PG6/TOD12A/SSLB0 pin.       |
|        |          |         |     | 00: PG6 input/output (port)                             |
|        |          |         |     | 01: TOD12A output (ATU-III)                             |
|        |          |         |     | 10: SSLB0 input/output (RSPI)                           |
|        |          |         |     | 11: Setting prohibited                                  |
| 11, 10 | PG5MD    | 00      | R/W | PG5 Mode  |
|        | [1:0]    |         |     | Control the function of the PG5/TOD11A/SSLA5/SSLC3 pin. |
|        |          |         |     | 00: PG5 input/output (port)                             |
|        |          |         |     | 01: TOD11A output (ATU-III)                             |
|        |          |         |     | 10: SSLA5 output (RSPI)                                 |
|        |          |         |     | 11: SSLC3 output (RSPI)                                 |

| Bit  | Bit Name | Initial<br>Value | R/W   | Description   |
|------|----------|------------------|-------|---|
| 9, 8 | PG4MD    | 00               | R/W   | PG4 Mode  |
| 3, 0 | [1:0]    | 00               | 10,44 | Control the function of the PG4/TOD10A/SSLA4/SSLB3 pin. |
|      |          |                  |       | 00: PG4 input/output (port)                             |
|      |          |                  |       | 01: TOD10A output (ATU-III)                             |
|      |          |                  |       | 10: SSLA4 output (RSPI)                                 |
|      |          |                  |       | 11: SSLB3 output (RSPI)                                 |
| 7, 6 | PG3MD    | 00               | R/W   | PG3 Mode  |
|      | [1:0]    |                  |       | Control the function of the PG3/TOD03A/SSLA3 pin.       |
|      |          |                  |       | 00: PG3 input/output (port)                             |
|      |          |                  |       | 01: TOD03A output (ATU-III)                             |
|      |          |                  |       | 10: SSLA3 output (RSPI)                                 |
|      |          |                  |       | 11: Setting prohibited                                  |
| 5, 4 | PG2MD    | 00               | R/W   | PG2 Mode  |
|      | [1:0]    |                  |       | Control the function of the PG2/TOD02A/SSLA2 pin.       |
|      |          |                  |       | 00: PG2 input/output (port)                             |
|      |          |                  |       | 01: TOD02A output (ATU-III)                             |
|      |          |                  |       | 10: SSLA2 output (RSPI)                                 |
|      |          |                  |       | 11: Setting prohibited                                  |
| 3, 2 | PG1MD    | 00               | R/W   | PG1 Mode  |
|      | [1:0]    |                  |       | Control the function of the PG1/TOD01A/SSLA1 pin.       |
|      |          |                  |       | 00: PG1 input/output (port)                             |
|      |          |                  |       | 01: TOD01A output (ATU-III)                             |
|      |          |                  |       | 10: SSLA1 output (RSPI)                                 |
|      |          |                  |       | 11: Setting prohibited                                  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description                                       |
|------|----------|------------------|-----|---|
| 1, 0 | PG0MD    | 00               | R/W | PG0 Mode  |
|      | [1:0]    |                  |     | Control the function of the PG0/TOD00A/SSLA0 pin. |
|      |          |                  |     | 00: PG0 input/output (port)                       |
|      |          |                  |     | 01: TOD00A output (ATU-III)                       |
|      |          |                  |     | 10: SSLA0 input/output (RSPI)                     |
|      |          |                  |     | 11: Setting prohibited                            |

### 22.1.15 Port H I/O Register (PHIOR)

PHIOR is a 16-bit readable/writable register that sets the I/O direction of the port H pins.

PHIOR is enabled only when the port H pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PHIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|----|----|----|----|----|----|---|---|---|---|------------|------------|------------|------------|------------|------------|
|                | -  | -  | -  | -  | -  | -  | - | - | - | - | PH5<br>IOR | PH4<br>IOR | PH3<br>IOR | PH2<br>IOR | PH1<br>IOR | PH0<br>IOR |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0          | 0          | 0          | 0          | 0          | 0          |
| B/W-           | P  | D  | D  | D  | D  | D  | D | D | D | D | D/M        | D/M        | D/M        | D/M        | D/M        | D/M        |

| Bit     | Bit Name            | Initial<br>Value | R/W | Description  |
|---------|---------------------|------------------|-----|--|
| 15 to 6 | _                   | All 0            | R   | Reserved   |
|         |                     |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 5 to 0  | PH5IOR to<br>PH0IOR | All 0            | R/W | The PH5IOR to PH0IOR bits correspond to the PH5 to PH0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|         |                     |                  |     | 0: The corresponding pin is set to input.  |
|         |                     |                  |     | 1: The corresponding pin is set to output.   |

# 22.1.16 Port H Control Register (PHCR)

PHCR is a 16-bit readable/writable register that controls the functions of multiplexed pins of port H.

PHCR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11   | 10      | 9    | 8       | 7    | 6       | 5    | 4       | 3    | 2       | 1    | 0       |
|----------------|----|----|----|----|------|---------|------|---------|------|---------|------|---------|------|---------|------|---------|
|                | -  | -  | -  | -  | PH5M | ID[1:0] | PH4N | ID[1:0] | PH3N | ID[1:0] | PH2N | ID[1:0] | PH1M | ID[1:0] | PH0M | ID[1:0] |
| Initial value: | 0  | 0  | 0  | 0  | 0    | 0       | 0    | 0       | 0    | 0       | 0    | 0       | 0    | 0       | 0    | 0       |
| R/W:           | R  | R  | R  | R  | R/W  | R/W     |

|          |          | Initial |     |  |
|----------|----------|---------|-----|--|
| Bit      | Bit Name | Value   | R/W | Description  |
| 15 to 12 | _        | All 0   | R   | Reserved   |
|          |          |         |     | These bits are always read as 0. The write value should always be 0. |
| 11, 10   | PH5MD    | 00      | R/W | PH5 Mode   |
|          | [1:0]    |         |     | Control the function of the PH5/TIF5 pin.                            |
|          |          |         |     | 00: PH5 input/output (port)  |
|          |          |         |     | 01: Setting prohibited   |
|          |          |         |     | 10: Setting prohibited   |
|          |          |         |     | 11: TIF5 input (ATU-III)   |
| 9, 8     | PH4MD    | 00      | R/W | PH4 Mode   |
|          | [1:0]    |         |     | Control the function of the PH4/TIF4 pin.                            |
|          |          |         |     | 00: PH4 input/output (port)  |
|          |          |         |     | 01: Setting prohibited   |
|          |          |         |     | 10: Setting prohibited   |
|          |          |         |     | 11: TIF4 input (ATU-III)   |
| 7, 6     | PH3MD    | 00      | R/W | PH3 Mode   |
|          | [1:0]    |         |     | Control the function of the PH3/TIF3 pin.                            |
|          |          |         |     | 00: PH3 input/output (port)  |
|          |          |         |     | 01: Setting prohibited   |
|          |          |         |     | 10: Setting prohibited   |
|          |          |         |     | 11: TIF3 input (ATU-III)   |
|          |          |         |     |  |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 5, 4 | PH2MD    | 00               | R/W | PH2 Mode   |
|      | [1:0]    |                  |     | Control the function of the PH2/TIF2A pin.         |
|      |          |                  |     | 00: PH2 input/output (port)                        |
|      |          |                  |     | 01: Setting prohibited                             |
|      |          |                  |     | 10: Setting prohibited                             |
|      |          |                  |     | 11: TIF2A input (ATU-III)                          |
| 3, 2 | PH1MD    | 00               | R/W | PH1 Mode   |
| [-   | [1:0]    |                  |     | Control the function of the PH1/ADTRG_B/TIF1A pin. |
|      |          |                  |     | 00: PH1 input/output (port)                        |
|      |          |                  |     | 01: Setting prohibited                             |
|      |          |                  |     | 10: ADTRG_B input (ADC)                            |
|      |          |                  |     | 11: TIF1A input (ATU-III)                          |
| 1, 0 | PH0MD    | 00               | R/W | PH0 Mode   |
|      | [1:0]    |                  |     | Control the function of the PH0/ADTRG_A/TIF0A pin. |
|      |          |                  |     | 00: PH0 input/output (port)                        |
|      |          |                  |     | 01: Setting prohibited                             |
|      |          |                  |     | 10: ADTRG_A input (ADC)                            |
|      |          |                  |     | 11: TIF0A input (ATU-III)                          |

### 22.1.17 Port J I/O Register (PJIOR)

PJIOR is a 16-bit readable/writable register that sets the I/O direction of the port J pins.

PJIOR is enabled only when the port J pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PJIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|----|----|----|----|----|----|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | -  | -  | -  | -  | -  | -  | PJ9<br>IOR | PJ8<br>IOR | PJ7<br>IOR | PJ6<br>IOR | PJ5<br>IOR | PJ4<br>IOR | PJ3<br>IOR | PJ2<br>IOR | PJ1<br>IOR | PJ0<br>IOR |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R  | R  | R  | R  | R  | R  | R/W        |

| Bit      | Bit Name            | Initial<br>Value | R/W | Description  |
|----------|---------------------|------------------|-----|--|
| 15 to 10 | _                   | All 0            | R   | Reserved   |
|          |                     |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 9 to 0   | PJ9IOR to<br>PJ0IOR | All 0            | R/W | The PJ9IOR to PJ0IOR bits correspond to the PJ9 to PJ0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|          |                     |                  |     | 0: The corresponding pin is set to input.  |
|          |                     |                  |     | 1: The corresponding pin is set to output.   |

# 22.1.18 Port J Control Registers 1 and 2 (PJCR1 and PJCR2)

PJCR1 and PJCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port J.

PJCR1 and PJCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

# (1) Port J Control Register 2 (PJCR2)

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2         | 1 | 0         |
|----------------|----|----|----|----|----|----|---|---|---|---|---|---|---|-----------|---|-----------|
|                | -  | -  | -  | -  | -  | -  | - | - | - | - | - | - | - | PJ9<br>MD | - | PJ8<br>MD |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0         | 0 | 0         |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R | R | R/W       | R | R/W       |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 15 to 3 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 2       | PJ9MD    | 0                | R/W | PJ9 Mode   |
|         |          |                  |     | Controls the function of the PJ9/RxD_B pin.                          |
|         |          |                  |     | 0: PJ9 input/output (port)   |
|         |          |                  |     | 1: RxD_B input (SCI)   |
| 1       | _        | 0                | R   | Reserved   |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0.    |
| 0       | PJ8MD    | 0                | R/W | PJ8 Mode   |
|         |          |                  |     | Controls the function of the PJ8/TxD_B pin.                          |
|         |          |                  |     | 0: PJ8 input/output (port)   |
|         |          |                  |     | 1: TxD_B output (SCI)  |

# (2) Port J Control Register 1 (PJCR1)

| Bit:           | 15   | 14     | 13 | 12        | 11 | 10        | 9    | 8      | 7    | 6       | 5    | 4       | 3    | 2       | 1    | 0      |  |
|----------------|------|--------|----|-----------|----|-----------|------|--------|------|---------|------|---------|------|---------|------|--------|--|
|                | PJ7M | D[1:0] | -  | PJ6<br>MD | -  | PJ5<br>MD | PJ4M | D[1:0] | PJ3N | ID[1:0] | PJ2N | ID[1:0] | PJ1M | ID[1:0] | PJ0M | D[1:0] |  |
| Initial value: | 0    | 0      | 0  | 0         | 0  | 0         | 0    | 0      | 0    | 0       | 0    | 0       | 0    | 0       | 0    | 0      |  |
| R/W:           | R/W  | R/W    | R  | 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    |  |

|        |          | Initial |     |   |
|--------|----------|---------|-----|---|
| Bit    | Bit Name | Value   | R/W | Description   |
| 15, 14 | PJ7MD    | 00      | R/W | PJ7 Mode  |
|        | [1:0]    |         |     | Control the function of the                                       |
|        |          |         |     | PJ7/SCK_B/ADEND_A/TIJ1 pin.                                       |
|        |          |         |     | 00: PJ7 input/output (port)                                       |
|        |          |         |     | 01: SCK_B input/output (SCI)                                      |
|        |          |         |     | 10: ADEND_A output (ADC)  |
|        |          |         |     | 11: TIJ1 input (ATU-III)  |
| 13     | _        | 0       | R   | Reserved  |
|        |          |         |     | This bit is always read as 0. The write value should always be 0. |
| 12     | PJ6MD    | 0       | R/W | PJ6 Mode  |
|        |          |         |     | Controls the function of the PJ6/RxD_A pin.                       |
|        |          |         |     | 0: PJ6 input/output (port)  |
|        |          |         |     | 1: RxD_A input (SCI)  |
| 11     | _        | 0       | R   | Reserved  |
|        |          |         |     | This bit is always read as 0. The write value should always be 0. |
| 10     | PJ5MD    | 0       | R/W | PJ5 Mode  |
|        |          |         |     | Controls the function of the PJ5/TxD_A pin.                       |
|        |          |         |     | 0: PJ5 input/output (port)  |
|        |          |         |     | 1: TxD_A output (SCI)   |

| Bit  | Bit Name | Initial<br>Value | R/W | Description  |
|------|----------|------------------|-----|--|
| 9, 8 | PJ4MD    | 00               | R/W | PJ4 Mode   |
| , -  | [1:0]    |                  |     | Control the function of the PJ4/SCK_A/ADEND_B/TIJ0 pin.            |
|      |          |                  |     | 00: PJ4 input/output (port)  |
|      |          |                  |     | 01: SCK_A input/output (SCI)                                       |
|      |          |                  |     | 10: ADEND_B output (ADC)   |
|      |          |                  |     | 11: TIJ0 input (ATU-III)   |
| 7, 6 | PJ3MD    | 00               | R/W | PJ3 Mode   |
|      | [1:0]    |                  |     | Control the function of the PJ3/RxD_A/CRx_C/CRx_A&CRx_B&CRx_C pin. |
|      |          |                  |     | 00: PJ3 input/output (port)  |
|      |          |                  |     | 01: RxD_A input (SCI)  |
|      |          |                  |     | 10: CRx_C input (RCAN-TL1)   |
|      |          |                  |     | 11: CRx_A&CRx_B&CRx_C input (RCAN-TL1)                             |
| 5, 4 | PJ2MD    | 00               | R/W | PJ2 Mode   |
|      | [1:0]    |                  |     | Control the function of the PJ2/TxD_A/CTx_C/CTx_A&CTx_B&CTx_C pin. |
|      |          |                  |     | 00: PJ2 input/output (port)  |
|      |          |                  |     | 01: TxD_A output (SCI)   |
|      |          |                  |     | 10: CTx_C output (RCAN-TL1)  |
|      |          |                  |     | 11: CTx_A&CTx_B&CTx_C output (RCAN-TL1)                            |
| 3, 2 | PJ1MD    | 00               | R/W | PJ1 Mode   |
|      | [1:0]    |                  |     | Control the function of the PJ1/RxD_A/CRx_A/CRx_A&CRx_B pin.       |
|      |          |                  |     | 00: PJ1 input/output (port)  |
|      |          |                  |     | 01: RxD_A input (SCI)  |
|      |          |                  |     | 10: CRx_A input (RCAN-TL1)   |
|      |          |                  |     | 11: CRx_A&CRx_B input (RCAN-TL1)                                   |

|      |          | Initial |     |  |
|------|----------|---------|-----|--|
| Bit  | Bit Name | Value   | R/W | Description  |
| 1, 0 | PJ0MD    | 00      | R/W | PJ0 Mode   |
| [:   | [1:0]    |         |     | Control the function of the PJ0/TxD_A/CTx_A/CTx_A&CTx_B pin. |
|      |          |         |     | 00: PJ0 input/output (port)                                  |
|      |          |         |     | 01: TxD_A output (SCI)                                       |
|      |          |         |     | 10: CTx_A output (RCAN-TL1)                                  |
|      |          |         |     | 11: CTx_A&CTx_B output (RCAN-TL1)                            |

# 22.1.19 Port K I/O Register (PKIOR)

PKIOR is a 16-bit readable/writable register that sets the I/O direction of the port K pins.

PKIOR is enabled only when the port K pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PKIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|----|----|----|----|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | -  | -  | -  | -  | PK11<br>IOR | PK10<br>IOR | PK9<br>IOR | PK8<br>IOR | PK7<br>IOR | PK6<br>IOR | PK5<br>IOR | PK4<br>IOR | PK3<br>IOR | PK2<br>IOR | PK1<br>IOR | PK0<br>IOR |
| Initial value: | 0  | 0  | 0  | 0  | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W·           | R  | R  | R  | R  | R/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<br>Value | R/W | Description  |
|----------|----------------------|------------------|-----|--|
| 15 to 12 | _                    | All 0            | R   | Reserved   |
|          |                      |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 11 to 0  | PK11IOR to<br>PK0IOR | All 0            | R/W | The PK11IOR to PK0IOR bits correspond to the PK11 to PK0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|          |                      |                  |     | 0: The corresponding pin is set to input.  |
|          |                      |                  |     | 1: The corresponding pin is set to output.   |

# 22.1.20 Port K Control Registers 1 and 2 (PKCR1 and PKCR2)

PKCR1 and PKCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port K.

PKCR1 and PKCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

## (1) Port K Control Register 2 (PKCR2)

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6       | 5     | 4       | 3    | 2       | 1 | 0         |
|----------------|----|----|----|----|----|----|---|---|-------|---------|-------|---------|------|---------|---|-----------|
|                | -  | -  | -  | -  | -  | -  | - | - | PK11N | ИD[1:0] | PK10N | ИD[1:0] | PK9M | ID[1:0] | - | PK8<br>MD |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0     | 0       | 0     | 0       | 0    | 0       | 0 | 0         |
| R/W·           | R  | R  | R  | R  | R  | R  | R | R | R/W   | R/W     | R/W   | R/W     | R/W  | R/W     | R | R/W       |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 15 to 8 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 7, 6    | PK11MD   | 00               | R/W | PK11 Mode  |
|         | [1:0]    |                  |     | Control the function of the PK11/MISOC pin.                          |
|         |          |                  |     | 00: PK11 input/output (port)   |
|         |          |                  |     | 01: Setting prohibited   |
|         |          |                  |     | 10: MISOC input/output (RSPI)  |
|         |          |                  |     | 11: Setting prohibited   |
| 5, 4    | PK10MD   | 00               | R/W | PK10 Mode  |
|         | [1:0]    |                  |     | Control the function of the PK10/MOSIC pin.                          |
|         |          |                  |     | 00: PK10 input/output (port)   |
|         |          |                  |     | 01: Setting prohibited   |
|         |          |                  |     | 10: MOSIC input/output (RSPI)  |
|         |          |                  |     | 11: Setting prohibited   |

| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 3, 2 | PK9MD    | 00               | R/W | PK9 Mode  |
|      | [1:0]    |                  |     | Control the function of the PK9/RSPCKC pin.                       |
|      |          |                  |     | 00: PK9 input/output (port)                                       |
|      |          |                  |     | 01: Setting prohibited  |
|      |          |                  |     | 10: RSPCKC input/output (RSPI)                                    |
|      |          |                  |     | 11: Setting prohibited  |
| 1    | _        | 0                | R   | Reserved  |
|      |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 0    | PK8MD    | 0                | R/W | PK8 Mode  |
|      |          |                  |     | Controls the function of the PK8/RxD_E pin.                       |
|      |          |                  |     | 0: PK8 input/output (port)  |
|      |          |                  |     | 1: RxD_E input (SCI)  |

# (2) Port K Control Register 1 (PKCR1)

| Bit:           | 15 | 14        | 13 | 12        | 11   | 10      | 9    | 8      | 7    | 6       | 5    | 4      | 3    | 2      | 1    | 0      |
|----------------|----|-----------|----|-----------|------|---------|------|--------|------|---------|------|--------|------|--------|------|--------|
|                | -  | PK7<br>MD | -  | PK6<br>MD | PK5M | ID[1:0] | PK4M | D[1:0] | PK3M | ID[1:0] | PK2M | D[1:0] | PK1M | D[1:0] | PK0M | D[1:0] |
| Initial value: | 0  | 0         | 0  | 0         | 0    | 0       | 0    | 0      | 0    | 0       | 0    | 0      | 0    | 0      | 0    | 0      |
| R/W:           | R  | R/W       | R  | R/W       | R/W  | R/W     | R/W  | R/W    | R/W  | R/W     | R/W  | R/W    | R/W  | R/W    | R/W  | R/W    |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 15     | _        | 0                | R   | Reserved  |
|        |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 14     | PK7MD    | 0                | R/W | PK7 Mode  |
|        |          |                  |     | Controls the function of the PK7/TxD_E pin.                       |
|        |          |                  |     | 0: PK7 input/output (port)  |
|        |          |                  |     | 1: TxD_E output (SCI)   |
| 13     | _        | 0                | R   | Reserved  |
|        |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 12     | PK6MD    | 0                | R/W | PK6 Mode  |
|        |          |                  |     | Controls the function of the PK6/SCK_E pin.                       |
|        |          |                  |     | 0: PK6 input/output (port)  |
|        |          |                  |     | 1: SCK_E input/output (SCI)                                       |
| 11, 10 | PK5MD    | 00               | R/W | PK5 Mode  |
|        | [1:0]    |                  |     | Control the function of the PK5/RxD_D/MISOB pin.                  |
|        |          |                  |     | 00: PK5 input/output (port)                                       |
|        |          |                  |     | 01: RxD_D input (SCI)   |
|        |          |                  |     | 10: MISOB input/output (RSPI)                                     |
|        |          |                  |     | 11: Setting prohibited  |
| 9, 8   | PK4MD    | 00               | R/W | PK4 Mode  |
|        | [1:0]    |                  |     | Control the function of the PK4/TxD_D/MOSIB pin.                  |
|        |          |                  |     | 00: PK4 input/output (port)                                       |
|        |          |                  |     | 01: TxD_D output (SCI)  |
|        |          |                  |     | 10: MOSIB input/output (RSPI)                                     |
| -      |          |                  |     | 11: Setting prohibited  |

|      |          | Initial |     |  |
|------|----------|---------|-----|--|
| Bit  | Bit Name | Value   | R/W | Description  |
| 7, 6 | PK3MD    | 00      | R/W | PK3 Mode   |
|      | [1:0]    |         |     | Control the function of the PK3/SCK_D/RSPCKB pin.        |
|      |          |         |     | 00: PK3 input/output (port)                              |
|      |          |         |     | 01: SCK_D input/output (SCI)                             |
|      |          |         |     | 10: RSPCKB input/output (RSPI)                           |
|      |          |         |     | 11: Setting prohibited                                   |
| 5, 4 | PK2MD    | 00      | R/W | PK2 Mode   |
|      | [1:0]    |         |     | Control the function of the PK2/RxD_C/MISOA pin.         |
|      |          |         |     | 00: PK2 input/output (port)                              |
|      |          |         |     | 01: RxD_C input (SCI)                                    |
|      |          |         |     | 10: MISOA input/output (RSPI)                            |
|      |          |         |     | 11: Setting prohibited                                   |
| 3, 2 | PK1MD    | 00      | R/W | PK1 Mode   |
|      | [1:0]    |         |     | Control the function of the PK1/TxD_C/MOSIA pin.         |
|      |          |         |     | 00: PK1 input/output (port)                              |
|      |          |         |     | 01: TxD_C output (SCI)                                   |
|      |          |         |     | 10: MOSIA input/output (RSPI)                            |
|      |          |         |     | 11: Setting prohibited                                   |
| 1, 0 | PK0MD    | 00      | R/W | PK0 Mode   |
|      | [1:0]    |         |     | Control the function of the PK0/SCK_C/RSPCKA/UBCTRG pin. |
|      |          |         |     | 00: PK0 input/output (port)                              |
|      |          |         |     | 01: SCK_C input/output (ATU-III)                         |
|      |          |         |     | 10: RSPCKA input/output (RSPI)                           |
|      |          |         |     | 11: UBCTRG output (UBC)                                  |

# 22.1.21 Port L I/O Register (PLIOR)

PLIOR is a 16-bit readable/writable register that sets the I/O direction of the port L pins.

PLIOR is enabled only when the port L pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PLIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|----|----|----|----|----|----|---|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | -  | -  | -  | -  | -  | -  | - | PL8<br>IOR | PL7<br>IOR | PL6<br>IOR | PL5<br>IOR | PL4<br>IOR | PL3<br>IOR | PL2<br>IOR | PL1<br>IOR | PL0<br>IOR |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R/W        |

| Bit     | Bit Name            | Initial<br>Value | R/W | Description  |
|---------|---------------------|------------------|-----|--|
| 15 to 9 | _                   | All 0            | R   | Reserved   |
|         |                     |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 8 to 0  | PL8IOR to<br>PL0IOR | All 0            | R/W | The PL8IOR to PL0IOR bits correspond to the PL8 to PL0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). |
|         |                     |                  |     | 0: The corresponding pin is set to input.  |
|         |                     |                  |     | 1: The corresponding pin is set to output.   |

### Port L Control Registers 1 and 2 (PLCR1 and PLCR2)

PLCR1 and PLCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port L.

PLCR1 and PLCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

#### Port L Control Register 2 (PLCR2) **(1)**

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0         |
|----------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-----------|
|                | -  | -  | -  | -  | -  | -  | - | - | - | - | - | - | - | - | - | PL8<br>MD |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0         |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R | R | R | R | R/W       |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 15 to 1 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 0       | PL8MD    | 0                | R/W | PL8 Mode   |
|         |          |                  |     | Controls the function of the PL8/TOE33 pin.                          |
|         |          |                  |     | 0: PL8 input/output (port)   |
|         |          |                  |     | 1: TOE33 output (ATU-III)  |

# (2) Port L Control Register 1 (PLCR1)

| Bit:           | 15   | 14     | 13   | 12     | 11   | 10     | 9    | 8      | 7    | 6       | 5    | 4      | 3    | 2      | 1    | 0      |
|----------------|------|--------|------|--------|------|--------|------|--------|------|---------|------|--------|------|--------|------|--------|
|                | PL7M | D[1:0] | PL6M | D[1:0] | PL5M | D[1:0] | PL4M | D[1:0] | PL3M | ID[1:0] | PL2M | D[1:0] | PL1M | D[1:0] | PLOM | D[1:0] |
| Initial value: | 0    | 0      | 0    | 0      | 0    | 0      | 0    | 0      | 0    | 0       | 0    | 0      | 0    | 0      | 0    | 0      |
| R/W:           | R/W  | R/W     | R/W  | R/W    | R/W  | R/W    | R/W  | R/W    |

| Bit    | Bit Name | Initial<br>Value | R/W | Description                                     |
|--------|----------|------------------|-----|---|
| 15, 14 | PL7MD    | 00               | R/W | PL7 Mode  |
|        | [1:0]    |                  |     | Control the function of the PL7/TOE32/IRQ7 pin. |
|        |          |                  |     | 00: PL7 input/output (port)                     |
|        |          |                  |     | 01: TOE32 output (ATU-III)                      |
|        |          |                  |     | 10: IRQ7 input (INTC)                           |
|        |          |                  |     | 11: Setting prohibited                          |
| 13, 12 | PL6MD    | 00               | R/W | PL6 Mode  |
|        | [1:0]    |                  |     | Control the function of the PL6/TOE31/IRQ6 pin. |
|        |          |                  |     | 00: PL6 input/output (port)                     |
|        |          |                  |     | 01: TOE31 output (ATU-III)                      |
|        |          |                  |     | 10: IRQ6 input (INTC)                           |
|        |          |                  |     | 11: Setting prohibited                          |
| 11, 10 | PL5MD    | 00               | R/W | PL5 Mode  |
|        | [1:0]    |                  |     | Control the function of the PL5/TOE30/IRQ5 pin. |
|        |          |                  |     | 00: PL5 input/output (port)                     |
|        |          |                  |     | 01: TOE30 output (ATU-III)                      |
|        |          |                  |     | 10: IRQ5 input (INTC)                           |
|        |          |                  |     | 11: Setting prohibited                          |
| 9, 8   | PL4MD    | 00               | R/W | PL4 Mode  |
|        | [1:0]    |                  |     | Control the function of the PL4/TOE23/IRQ4 pin. |
|        |          |                  |     | 00: PL4 input/output (port)                     |
|        |          |                  |     | 01: TOE23 output (ATU-III)                      |
|        |          |                  |     | 10: IRQ4 input (INTC)                           |
|        |          |                  |     | 11: Setting prohibited                          |

| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 7, 6 | PL3MD    | 00               | R/W | PL3 Mode  |
|      | [1:0]    |                  |     | Control the function of the PL3/TOE22/IRQ3 pin.         |
|      |          |                  |     | 00: PL3 input/output (port)                             |
|      |          |                  |     | 01: TOE22 output (ATU-III)                              |
|      |          |                  |     | 10: IRQ3 input (INTC)                                   |
|      |          |                  |     | 11: Setting prohibited                                  |
| 5, 4 | PL2MD    | 00               | R/W | PL2 Mode  |
|      | [1:0]    |                  |     | Control the function of the PL2/TOE21/IRQ2 pin.         |
|      |          |                  |     | 00: PL2 input/output (port)                             |
|      |          |                  |     | 01: TOE21 output (ATU-III)                              |
|      |          |                  |     | 10: IRQ2 input (INTC)                                   |
|      |          |                  |     | 11: Setting prohibited                                  |
| 3, 2 | PL1MD    | 00               | R/W | PL1 Mode  |
|      | [1:0]    |                  |     | Control the function of the PL1/TOE20/IRQ1/POD pin.     |
|      |          |                  |     | 00: PL1 input/output (port)                             |
|      |          |                  |     | 01: TOE20 output (ATU-III)                              |
|      |          |                  |     | 10: IRQ1 input (INTC)                                   |
|      |          |                  |     | 11: POD input (port)                                    |
| 1, 0 | PL0MD    | 00               | R/W | PL0 Mode  |
|      | [1:0]    |                  |     | Control the function of the PL0/ $\overline{IRQ0}$ pin. |
|      |          |                  |     | 00: PL0 input/output (port)                             |
|      |          |                  |     | 01: Setting prohibited                                  |
|      |          |                  |     | 10: IRQ0 input (INTC)                                   |
|      |          |                  |     | 11: Setting prohibited                                  |

# Section 23 I/O Ports

### 23.1 Overview

This LSI provides 11 ports: ports A, B, C, D, E, F, G, H, J, K, and L.

Ports A, C, F, and G are 16-bit I/O ports. Port B is a 15-bit I/O port. Ports D and E are 14-bit I/O ports. Port H is a 6-bit I/O port. Port J is a 10-bit I/O port. Port K is a 12-bit I/O port. Port L is a 9-bit I/O port.

Each port pin which is a general I/O port also has other multiplexed functions. The function of the multiplexed pin is selected by the pin function controller (PFC).

Each port has a data register to store pin data and a port register to read the pin state.

In addition, each of the ports B, D, E, F, G, J, K, and L incorporates an inverting register to output the inverted set value.

Port G can detect an edge input on a pin. Port G has an edge selecting register to select an edge.

Ports B, E, F, G, J, and K can set the driving ability of pins. Each of these ports has a driving ability setting register for this purpose.

Ports B, F, J, and K can set the pull-down for pins. Each of these ports has a pin state setting register for this purpose.

In addition, this LSI provides the CK control register (CKCR) to enable or disable the CK pin output.

# 23.2 Register Descriptions

The I/O port registers are listed in table 23.1.

**Table 23.1 Register Configuration** 

| Register Name                           | Abbreviation | R/W | Initial Value | Address    | Access<br>Size |
|---|--------------|-----|---------------|------------|----------------|
| Port A data register                    | PADR         | R/W | H'0000        | H'FFFE3802 | 8, 16          |
| Port A port register                    | PAPR         | R   | Pin state     | H'FFFE381E | 8, 16          |
| Port B data register                    | PBDR         | R/W | H'0000        | H'FFFE3882 | 8, 16          |
| Port B port register                    | PBPR         | R   | Pin state     | H'FFFE389E | 8, 16          |
| Port B inverting register               | PBIR         | R/W | H'0000        | H'FFFE3898 | 8, 16, 32      |
| Port B driving ability setting register | PBDSR        | R/W | H'0000        | H'FFFE389A | 8, 16          |
| Port B pin state setting register       | PBPSR        | R/W | H'0000        | H'FFFE389C | 8, 16, 32      |
| Port C data register                    | PCDR         | R/W | H'0000        | H'FFFE3902 | 8, 16          |
| Port C port register                    | PCPR         | R   | Pin state     | H'FFFE391E | 8, 16          |
| Port D data register                    | PDDR         | R/W | H'0000        | H'FFFFC800 | 8, 16, 32      |
| Port D port register                    | PDPR         | R   | Pin state     | H'FFFFC802 | 8, 16          |
| Port D inverting register               | PDIR         | R/W | H'0000        | H'FFFFC804 | 8, 16          |
| Port E data register                    | PEDR         | R/W | H'0000        | H'FFFFC810 | 8, 16, 32      |
| Port E port register                    | PEPR         | R   | Pin state     | H'FFFFC812 | 8, 16          |
| Port E inverting register               | PEIR         | R/W | H'0000        | H'FFFFC814 | 8, 16, 32      |
| Port E driving ability setting register | PEDSR        | R/W | H'0000        | H'FFFFC816 | 8, 16          |
| Port F data register                    | PFDR         | R/W | H'0000        | H'FFFFC820 | 8, 16, 32      |
| Port F port register                    | PFPR         | R   | Pin state     | H'FFFFC822 | 8, 16          |
| Port F inverting register               | PFIR         | R/W | H'0000        | H'FFFFC824 | 8, 16, 32      |
| Port F driving ability setting register | PFDSR        | R/W | H'0000        | H'FFFFC826 | 8, 16          |
| Port F pin state setting register       | PFPSR        | R/W | H'0000        | H'FFFFC828 | 8, 16, 32      |
| Port G data register                    | PGDR         | R/W | H'0000        | H'FFFFC830 | 8, 16, 32      |
| Port G port register                    | PGPR         | R   | Pin state     | H'FFFFC832 | 8, 16          |
| Port G inverting register               | PGIR         | R/W | H'0000        | H'FFFFC834 | 8, 16, 32      |

| Register Name                           | Abbreviation | R/W | Initial Value | Address    | Access<br>Size |
|---|--------------|-----|---------------|------------|----------------|
| Port G driving ability setting register | PGDSR        | R/W | H'0000        | H'FFFFC836 | 8, 16          |
| Port G edge selecting register          | PGER         | R/W | H'0000        | H'FFFFC838 | 8, 16          |
| Port H data register                    | PHDR         | R/W | H'0000        | H'FFFFC850 | 8, 16, 32      |
| Port H port register                    | PHPR         | R   | Pin state     | H'FFFFC852 | 8, 16          |
| Port J data register                    | PJDR         | R/W | H'0000        | H'FFFFC860 | 8, 16, 32      |
| Port J port register                    | PJPR         | R   | Pin state     | H'FFFFC862 | 8, 16          |
| Port J inverting register               | PJIR         | R/W | H'0000        | H'FFFFC864 | 8, 16, 32      |
| Port J driving ability setting register | PJDSR        | R/W | H'0000        | H'FFFFC866 | 8, 16          |
| Port J pin state setting register       | PJPSR        | R/W | H'0000        | H'FFFFC868 | 8, 16          |
| Port K data register                    | PKDR         | R/W | H'0000        | H'FFFFC880 | 8, 16, 32      |
| Port K port register                    | PKPR         | R   | Pin state     | H'FFFFC882 | 8, 16          |
| Port K inverting register               | PKIR         | R/W | H'0000        | H'FFFFC884 | 8, 16, 32      |
| Port K driving ability setting register | PKDSR        | R/W | H'0000        | H'FFFFC886 | 8, 16          |
| Port K pin state setting register       | PKPSR        | R/W | H'0000        | H'FFFFC888 | 8, 16          |
| Port L data register                    | PLDR         | R/W | H'0000        | H'FFFFC8A0 | 8, 16, 32      |
| Port L port register                    | PLPR         | R   | Pin state     | H'FFFFC8A2 | 8, 16          |
| Port L inverting register               | PLIR         | R/W | H'0000        | H'FFFFC8A4 | 8, 16          |
| CK control register                     | CKCR         | R/W | H'0000        | H'FFFFC920 | 8, 16          |

### 23.3 Port A

Port A is an input/output port with the 16 pins shown in figure 23.1.

|      |        | ROM disabled extension mode | ROM enabled extension mode | Single-chip mode |
|------|--------|-----------------------------|----------------------------|------------------|
|      | ][     | A0 (output)                 | PA0 (I/O)/A0 (output)      | PA0 (I/O)        |
|      |        | A1 (output)                 | PA1 (I/O)/A1 (output)      | PA1 (I/O)        |
|      |        | A2 (output)                 | PA2 (I/O)/A2 (output)      | PA2 (I/O)        |
|      | ←→[    | A3 (output)                 | PA3 (I/O)/A3 (output)      | PA3 (I/O)        |
|      | [      | A4 (output)                 | PA4 (I/O)/A4 (output)      | PA4 (I/O)        |
|      | ←→[    | A5 (output)                 | PA5 (I/O)/A5 (output)      | PA5 (I/O)        |
| I∢   | →→[    | A6 (output)                 | PA6 (I/O)/A6 (output)      | PA6 (I/O)        |
| Port | →→[    | A7 (output)                 | PA7 (I/O)/A7 (output)      | PA7 (I/O)        |
| 1 -  | ◆→[    | A8 (output)                 | PA8 (I/O)/A8 (output)      | PA8 (I/O)        |
|      |        | A9 (output)                 | PA9 (I/O)/A9 (output)      | PA9 (I/O)        |
|      |        | A10 (output)                | PA10 (I/O)/A10 (output)    | PA10 (I/O)       |
|      |        | A11 (output)                | PA11 (I/O)/A11 (output)    | PA11 (I/O)       |
|      |        | A12 (output)                | PA12 (I/O)/A12 (output)    | PA12 (I/O)       |
|      |        | A13 (output)                | PA13 (I/O)/A13 (output)    | PA13 (I/O)       |
|      |        | A14 (output)                | PA14 (I/O)/A14 (output)    | PA14 (I/O)       |
|      | ] , ر[ | A15 (output)                | PA15 (I/O)/A15 (output)    | PA15 (I/O)       |

Figure 23.1 Port A

### 23.3.1 Port A Data Register (PADR)

PADR is a 16-bit readable/writable register that stores port A data. The PA15DR to PA0DR bits correspond to the PA15/A15 to PA0/A0 pins, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.2 summarizes the PADR read and write operations.

PADR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15         | 14         | 13         | 12         | 11         | 10           | 9     | 8            | 7     | 6     | 5     | 4            | 3     | 2            | 1     | 0     |
|----------------|------------|------------|------------|------------|------------|--------------|-------|--------------|-------|-------|-------|--------------|-------|--------------|-------|-------|
|                | PA15<br>DR | PA14<br>DR | PA13<br>DR | PA12<br>DR | PA11<br>DR | PA10<br>DR   | PA9DR | PA8DR        | PA7DR | PA6DR | PA5DR | PA4DR        | PA3DR | PA2DR        | PA1DR | PA0DR |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0            | 0     | 0            | 0     | 0     | 0     | 0            | 0     | 0            | 0     | 0     |
| D/M.           | D/M        | D/M        | D/\//      | D/M        | D/M        | $D\Lambda M$ | D/M   | $D\Lambda M$ | D/\\/ | DAM   | D/M   | $D\Lambda M$ | R/M   | $D\Lambda M$ | D/M   | DΛΛ   |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions         |
|-----|----------|------------------|-----|----------------------|
|     |          |                  |     |                      |
| 15  | PA15DR   | 0                | R/W | Refer to table 23.2. |
| 14  | PA14DR   | 0                | R/W |                      |
| 13  | PA13DR   | 0                | R/W | _                    |
| 12  | PA12DR   | 0                | R/W | <del>-</del>         |
| 11  | PA11DR   | 0                | R/W | _                    |
| 10  | PA10DR   | 0                | R/W | _                    |
| 9   | PA9DR    | 0                | R/W | _                    |
| 8   | PA8DR    | 0                | R/W | _                    |
| 7   | PA7DR    | 0                | R/W | _                    |
| 6   | PA6DR    | 0                | R/W | _                    |
| 5   | PA5DR    | 0                | R/W | _                    |
| 4   | PA4DR    | 0                | R/W | _                    |
| 3   | PA3DR    | 0                | R/W | <del>-</del>         |
| 2   | PA2DR    | 0                | R/W | _                    |
| 1   | PA1DR    | 0                | R/W | _                    |
| 0   | PA0DR    | 0                | R/W | _                    |

# Table 23.2 Read and Write Operations of Port A Data Register (PADR)

• Bits 15 to 0 in the PADR register

| PAIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | Pin state  | Data can be written to PADR but data does not affect the pin state. |
|       | Other than general input  | Pin state  | Data can be written to PADR but data does not affect the pin state. |
| 1     | General output            | PADR value | Data written to PADR is output on the pin.                          |
| _     | Other than general output | PADR value | Data can be written to PADR but data does not affect the pin state. |

# 23.3.2 Port A Port Register (PAPR)

PAPR is a 16-bit read-only register that always stores port A pin states. This register cannot be directly written by the CPU. The PA15PR to PA0PR bits correspond to the PA15/A15 to PA0/A0 pins, respectively.

When PAPR is read, the pin state can be read.

| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|------------|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | PA15<br>PR | PA14<br>PR | PA13<br>PR | PA12<br>PR | PA11<br>PR | PA10<br>PR | PA9PR | PA8PR | PA7PR | PA6PR | PA5PR | PA4PR | PA3PR | PA2PR | PA1PR | PA0PR |
| Initial value: | PA15       | PA14       | PA13       | PA12       | PA11       | PA10       | PA9   | PA8   | PA7   | PA6   | PA5   | PA4   | PA3   | PA2   | PA1   | PA0   |
| R/W:           | R          | R          | R          | R          | R          | R          | R     | R     | R     | R     | R     | R     | R     | R     | R     | R     |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 15  | PA15PR   | Pin state        | R   | When these bits are read, the pin states can be read. |
| 14  | PA14PR   | Pin state        | R   | These bits cannot be modified.                        |
| 13  | PA13PR   | Pin state        | R   | -   |
| 12  | PA12PR   | Pin state        | R   | -   |
| 11  | PA11PR   | Pin state        | R   | <del>-</del>  |
| 10  | PA10PR   | Pin state        | R   | <del>-</del>  |
| 9   | PA9PR    | Pin state        | R   | -   |
| 8   | PA8PR    | Pin state        | R   | <del>-</del>  |
| 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   | <del>-</del>  |
| 1   | PA1PR    | Pin state        | R   | <del>-</del>  |
| 0   | PA0PR    | Pin state        | R   | _   |

### 23.4 Port B

Port B is an input/output port with the 15 pins shown in figure 23.2.

|        |             | (Area 0: 8 bits) (Area 0: 16 bits)  | ROM enabled extension mode                         | Single-chip mode                      |
|--------|-------------|-------------------------------------|--|---------------------------------------|
| _      | ]←→[        | A16 (output)                        | PB0 (I/O)/A16 (output)/MOSIA (I/O)                 | PB0 (I/O)/MOSIA (I/O)                 |
|        |             | A17 (output)                        | PB1 (I/O)/A17 (output)/MISOA (I/O)                 | PB1 (I/O)/MISOA (I/O)                 |
|        |             | A18 (output)                        | PB2 (I/O)/A18 (output)/MOSIB (I/O)                 | PB2 (I/O)/MOSIB (I/O)                 |
|        |             | A19 (output)                        | PB3 (I/O)/A19 (output)/MISOB (I/O)                 | PB3 (I/O)/MISOB (I/O)                 |
|        | <b> </b> →→ | A20 (output)                        | PB4 (I/O)/A20 (output)/CTx_B (output)/TIF6 (input) | PB4 (I/O)/CTx_B (output)/TIF6 (input) |
|        |             | A21 (output)                        | PB5 (I/O)/A21 (output)/CRx_B (input)/TIF7 (input)  | PB5 (I/O)/CRx_B (input)/TIF7 (input)  |
| В      |             | WE0 (output)                        | PB6 (I/O)/WE0 (output)                             | PB6 (I/O)                             |
| Port E | <b> </b> →→ | PB7 (I/O)/WE1 (output) WE1 (output) | PB7 (I/O)/WE1 (output)                             | PB7 (I/O)                             |
| ď      |             | WAIT (input)                        | PB8 (I/O)/WAIT (input)/TOE20 (output)              | PB8 (I/O)/TOE20 (output)              |
|        |             | RD (output)                         | PB9 (I/O)/RD (output)                              | PB9 (I/O)                             |
|        |             | CS0 (output)                        | PB10 (I/O)/CS0 (output)                            | PB10 (I/O)                            |
|        |             | PB11 (I/O)/ <del>CS1</del> (out     |  | PB11 (I/O)/TOE21 (output)             |
|        |             | PB12 (I/O)/CS2 (ou                  | tput)/RSPCKA (I/O)                                 | PB12 (I/O)/RSPCKA (I/O)               |
|        |             | PB13 (I/O)/CS3 (ou                  | tput)/RSPCKB (I/O)                                 | PB13 (I/O)/RSPCKB (I/O)               |
|        |             | RD/WR (output)                      | PB14 (I/O)/RD/WR (output)                          | PB14 (I/O)                            |

Figure 23.2 Port B

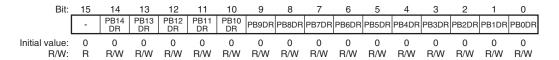
### 23.4.1 Port B Data Register (PBDR)

PBDR is a 16-bit readable/writable register that stores port B data. The PB14DR to PB0DR bits correspond to the PB14/RD/WR to PB0/A16/MOSIA pins, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.3 summarizes the PBDR read and write operations.

PBDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.



| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
|     | Dit Name |                  |     |   |
| 15  | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 14  | PB14DR   | 0                | R/W | Refer to table 23.3.  |
| 13  | PB13DR   | 0                | R/W | <del>-</del>  |
| 12  | PB12DR   | 0                | R/W | _   |
| 11  | PB11DR   | 0                | R/W | _   |
| 10  | PB10DR   | 0                | R/W | _   |
| 9   | PB9DR    | 0                | R/W | _   |
| 8   | PB8DR    | 0                | R/W | _   |
| 7   | PB7DR    | 0                | R/W | _   |
| 6   | PB6DR    | 0                | R/W | _   |
| 5   | PB5DR    | 0                | R/W | _   |
| 4   | PB4DR    | 0                | R/W | _   |
| 3   | PB3DR    | 0                | R/W | _   |
| 2   | PB2DR    | 0                | R/W | _   |
| 1   | PB1DR    | 0                | R/W | _   |
| 0   | PB0DR    | 0                | R/W | _   |

Table 23.3 Read and Write Operations of Port B Data Register (PBDR)

### • Bits 15 to 0 in PBDR

| PBIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | Pin state  | Data can be written to PBDR but data does not affect the pin state. |
|       | Other than general input  | Pin state  | Data can be written to PBDR but data does not affect the pin state. |
| 1     | General output            | PBDR value | Data written to PBDR is output on the pin.                          |
|       | Other than general output | PBDR value | Data can be written to PBDR but data does not affect the pin state. |

# 23.4.2 Port B Port Register (PBPR)

PBPR is a 16-bit read-only register that always stores port B pin states. This register cannot be directly written by the CPU. The PB14PR to PB0PR bits correspond to the PB14/RD/WR to PB0/A16/MOSIA pins, respectively.

When PBPR is read, the pin state can be read.

| Bit:           | 15 | 14         | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | -  | PB14<br>PR | PB13<br>PR | PB12<br>PR | PB11<br>PR | PB10<br>PR | PB9PR | PB8PR | PB7PR | PB6PR | PB5PR | PB4PR | PB3PR | PB2PR | PB1PR | PB0PR |
| Initial value: | 0  | PB14       | PB13       | PB12       | PB11       | PB10       | PB9   | PB8   | PB7   | PB6   | PB5   | PB4   | PB3   | PB2   | PB1   | PB0   |
| R/W:           | R  | R          | R          | R          | R          | R          | R     | R     | R     | R     | R     | R     | R     | R     | R     | R     |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 15  | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 14  | PB14PR   | Pin state        | R   | When these bits are read, the pin states can be read.             |
| 13  | PB13PR   | Pin state        | R   | These bits cannot be modified.                                    |
| 12  | PB12PR   | Pin state        | R   | _   |
| 11  | PB11PR   | Pin state        | R   | _   |
| 10  | PB10PR   | Pin state        | R   | _   |
| 9   | PB9PR    | Pin state        | R   | _   |
| 8   | PB8PR    | Pin state        | R   | _   |
| 7   | PB7PR    | Pin state        | R   | _   |
| 6   | PB6PR    | Pin state        | R   | _   |
| 5   | PB5PR    | Pin state        | R   | _   |
| 4   | PB4PR    | Pin state        | R   | _   |
| 3   | PB3PR    | Pin state        | R   | _   |
| 2   | PB2PR    | Pin state        | R   | _   |
| 1   | PB1PR    | Pin state        | R   | _   |
| 0   | PB0PR    | Pin state        | R   | <del>-</del>  |

# 23.4.3 Port B Inverting Register (PBIR)

PBIR is a 16-bit readable/writable register that enables or disables the inverting function for port B. Bits PB13IR to PB11IR, PB8IR, and PB4IR to PB0IR correspond to pins PB13/CS3/RSPCKB to PB11/CS1/TOE21, PB8/WAIT/TOE20, and PB4/A20/CTx\_B/TIF6 to PB0/A16/MOSIA, respectively. The PBIR setting is ignored when the corresponding pin function is specified as a bus function (A16 to A20, WAIT, or CS1 to CS3).

Setting the specific bits of PBIR to 1 reverts the output values on their corresponding pins.

PBIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13     | 12     | 11     | 10 | 9 | 8     | 7 | 6 | 5 | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|--------|--------|--------|----|---|-------|---|---|---|-------|-------|-------|-------|-------|
|                | -  | -  | PB13IR | PB12IR | PB11IR | -  | - | PB8IR | - | - | - | PB4IR | PB3IR | PB2IR | PB1IR | PB0IR |
| Initial value: | 0  | 0  | 0      | 0      | 0      | 0  | 0 | 0     | 0 | 0 | 0 | 0     | 0     | 0     | 0     | 0     |
| R/W·           | R  | R  | R/W    | R/W    | R/W    | R  | R | R/W   | R | R | R | R/W   | R/W   | R/W   | R/W   | R/W   |

| Bit    | Bit Name | Initial<br>Value | R/W | Descriptions  |  |  |  |
|--------|----------|------------------|-----|---|--|--|--|
| 15, 14 | _        | All 0            | R   | Reserved  |  |  |  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.                              |  |  |  |
| 13     | PB13IR   | 0                | R/W | These bits set whether or not the set value is inverted   |  |  |  |
| 12     | PB12IR   | 0                | R/W | when it is output on a corresponding pin.   |  |  |  |
| 11     | PB11IR   | 0                | R/W | 0: The set value is not inverted when output  |  |  |  |
|        |          |                  |     | 1: The set value is inverted when output  |  |  |  |
| 10, 9  | _        | All 0            | R   | Reserved  |  |  |  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.                              |  |  |  |
| 8      | PB8IR    | 0                | R/W | These bits set whether or not the set value is inverted when it is output on a corresponding pin. |  |  |  |
|        |          |                  |     | 0: The set value is not inverted when output  |  |  |  |
|        |          |                  |     | 1: The set value is inverted when output  |  |  |  |
| 7 to 5 | _        | All 0            | R   | Reserved  |  |  |  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.                              |  |  |  |
|        |          |                  |     |   |  |  |  |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 4   | PB4IR    | 0                | R/W | These bits set whether or not the set value is inverted |
| 3   | PB3IR    | 0                | R/W | when it is output on a corresponding pin.               |
| 2   | PB2IR    | 0                | R/W | ─ 0: The set value is not inverted when output          |
| 1   | PB1IR    | 0                | R/W | -1: The set value is inverted when output               |
| 0   | PB0IR    | 0                | R/W | _   |

# 23.4.4 Port B Driving Ability Setting Register (PBDSR)

PBDSR is a 16-bit readable/writable register that selects the driving ability for port B. Bits PB13DSR, PB12DSR, and PB4DSR to PB0DSR correspond to pins PB13/\overline{CS3}/RSPCKB, PB12/\overline{CS2}/RSPCKA, and PB4/A20/CTx\_B/TIF6 to PB0/A16/MOSIA, respectively. The PBDSR setting is ignored when the corresponding pin function is specified as bus function (A16 to A20, \overline{CS2}, and \overline{CS3}).

Setting the specific bits of PBDSR to 1 increases the driving ability of their corresponding pins.

PBDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13          | 12          | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4          | 3          | 2          | 1          | 0          |
|----------------|----|----|-------------|-------------|----|----|---|---|---|---|---|------------|------------|------------|------------|------------|
|                | -  | -  | PB13<br>DSR | PB12<br>DSR | -  | -  | - | - | - | - | 1 | PB4<br>DSR | PB3<br>DSR | PB2<br>DSR | PB1<br>DSR | PB0<br>DSR |
| Initial value: | 0  | 0  | 0           | 0           | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0          | 0          | 0          | 0          | 0          |
| R/W·           | R  | R  | R/W         | R/W         | R  | R  | R | R | R | R | R | R/W        | R/W        | R/W        | R/W        | R/W        |

| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions  |
|---------|----------|------------------|-----|---|
| 15, 14  | _        | All 0            | R   | Reserved  |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0.                              |
| 13      | PB13DSR  | 0                | R/W | When these bits are set to 1, the driving ability of the  |
| 12      | PB12DSR  | 0                | R/W | corresponding pin is higher than normal.  |
|         |          |                  |     | 0: Normal driving ability (slow slew rate)  |
|         |          |                  |     | 1: High driving ability (fast slew rate)  |
|         |          |                  |     | Note: For the characteristics of the driving ability, refer to section 32.3.14, Output Slew Rate. |
| 11 to 5 | _        | All 0            | R   | Reserved  |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0.                              |
| 4       | PB4DSR   | 0                | R/W | When these bits are set to 1, the driving ability of the  |
| 3       | PB3DSR   | 0                | R/W | corresponding pin is higher than normal.  |
| 2       | PB2DSR   | 0                | R/W | <ul><li>O: Normal driving ability (slow slew rate)</li></ul>                                      |
| 1       | PB1DSR   | 0                | R/W | -1: High driving ability (fast slew rate)   |
| ·       |          |                  | -   | - Note: For the characteristics of the driving ability, refer                                     |
| 0       | PB0DSR   | 0                | R/W | to section 32.3.14, Output Slew Rate.   |

### 23.4.5 Port B Pin State Setting Register (PBPSR)

PBPSR is a 16-bit readable/writable register that enables or disables the pull-down resistor of port B. Bits PB3PSR and PB1PSR correspond to pins PB3/A19/MISOB and PB1/A17/MISOA, respectively. The PBPSR setting is ignored when the corresponding pin function is specified as a bus function (A19 and A17).

Setting the specific bits of PBPSR to 1 puts their corresponding pins into pull-down state.

PBPSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3          | 2 | 1          | 0 |
|----------------|----|----|----|----|----|----|---|---|---|---|---|---|------------|---|------------|---|
|                | -1 | -  | -  | -  | -  | -  | - | - | - | - | - | - | PB3<br>PSR | - | PB1<br>PSR | - |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0          | 0 | 0          | 0 |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R | R | R | R | R | R/W        | R | R/W        | R |

| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions   |
|---------|----------|------------------|-----|--|
| 15 to 4 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0.             |
| 3       | PB3PSR   | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|         |          |                  |     | 0: Pull-down resistor disabled   |
|         |          |                  |     | 1: Pull-down resistor enabled  |
| 2       | _        | 0                | R   | Reserved   |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0.                |
| 1       | PB1PSR   | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|         |          |                  |     | 0: Pull-down resistor disabled   |
|         |          |                  |     | 1: Pull-down resistor enabled  |
| 0       | _        | 0                | R   | Reserved   |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0.                |

### 23.5 Port C

Port C is an input/output port with the 16 pins shown in figure 23.3.

|        |              | ROM disabled extended (Area 0: 8 bits) | ension mode<br>(Area 0: 16 bits) | ROM enabled<br>extension mode | Single-chip mode |
|--------|--------------|--|----------------------------------|-------------------------------|------------------|
|        | [            | D0 (                                   | [I/O)                            | PC0 (I/O)/D0 (I/O)            | PC0 (I/O)        |
|        |              | D1 (                                   | [I/O)                            | PC1 (I/O)/D1 (I/O)            | PC1 (I/O)        |
|        | <b> </b> → ▶ | D2 (                                   | [I/O)                            | PC2 (I/O)/D2 (I/O)            | PC2 (I/O)        |
|        | <b> </b> →→  | D3 (                                   | [I/O)                            | PC3 (I/O)/D3 (I/O)            | PC3 (I/O)        |
|        | <b> </b> →   | D4 (                                   | I/O)                             | PC4 (I/O)/D4 (I/O)            | PC4 (I/O)        |
|        | →→           | D5 (                                   | I/O)                             | PC5 (I/O)/D5 (I/O)            | PC5 (I/O)        |
| O      | →            | D6 (                                   | [I/O)                            | PC6 (I/O)/D6 (I/O)            | PC6 (I/O)        |
| Port ( | →            | D7 (                                   | [I/O)                            | PC7 (I/O)/D7 (I/O)            | PC7 (I/O)        |
| ď      |              | PC8 (I/O)/D8 (I/O)                     | D8 (I/O)                         | PC8 (I/O)/D8 (I/O)            | PC8 (I/O)        |
|        | ←→           | PC9 (I/O)/D9 (I/O)                     | D9 (I/O)                         | PC9 (I/O)/D9 (I/O)            | PC9 (I/O)        |
|        | ←→           | PC10 (I/O)/D10 (I/O)                   | D10 (I/O)                        | PC10 (I/O)/D10 (I/O)          | PC10 (I/O)       |
|        |              | PC11 (I/O)/D11 (I/O)                   | D11 (I/O)                        | PC11 (I/O)/D11 (I/O)          | PC11 (I/O)       |
|        |              | PC12 (I/O)/D12 (I/O)                   | D12 (I/O)                        | PC12 (I/O)/D12 (I/O)          | PC12 (I/O)       |
|        |              | PC13 (I/O)/D13 (I/O)                   | D13 (I/O)                        | PC13 (I/O)/D13 (I/O)          | PC13 (I/O)       |
|        | →            | PC14 (I/O)/D14 (I/O)                   | D14 (I/O)                        | PC14 (I/O)/D14 (I/O)          | PC14 (I/O)       |
|        |              | PC15 (I/O)/D15 (I/O)                   | D15 (I/O)                        | PC15 (I/O)/D15 (I/O)          | PC15 (I/O)       |

Figure 23.3 Port C

# 23.5.1 Port C Data Register (PCDR)

PCDR is a 16-bit readable/writable register that stores port C data. Bits PC15DR to PC0DR correspond to pins PC15/D15 to PC0/D0, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.4 summarizes the PCDR read and write operations.

PCDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|------------|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | PC15<br>DR | PC14<br>DR | PC13<br>DR | PC12<br>DR | PC11<br>DR | PC10<br>DR | PC9DR | PC8DR | PC7DR | PC6DR | PC5DR | PC4DR | PC3DR | PC2DR | PC1DR | PC0DR |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/M⋅           | D/\//      | D/M        | D/M        | D/M        | R/M        | D/M        | D/M   | D/M   | D/M   | D/\/  | D/M   | D/\// | D/M   | D/M   | D/\// | D/W   |

| D:  | Dit Name | Initial | DAM | Paradiations         |
|-----|----------|---------|-----|----------------------|
| Bit | Bit Name | Value   | R/W | Descriptions         |
| 15  | PC15DR   | 0       | R/W | Refer to table 23.4. |
| 14  | PC14DR   | 0       | R/W | _                    |
| 13  | PC13DR   | 0       | R/W | _                    |
| 12  | PC12DR   | 0       | R/W | <del>-</del>         |
| 11  | PC11DR   | 0       | R/W | _                    |
| 10  | PC10DR   | 0       | R/W | _                    |
| 9   | PC9DR    | 0       | R/W | _                    |
| 8   | PC8DR    | 0       | R/W | _                    |
| 7   | PC7DR    | 0       | R/W | <del>-</del>         |
| 6   | PC6DR    | 0       | R/W | <del>-</del>         |
| 5   | PC5DR    | 0       | R/W | _                    |
| 4   | PC4DR    | 0       | R/W | <del>-</del>         |
| 3   | PC3DR    | 0       | R/W | _                    |
| 2   | PC2DR    | 0       | R/W | _                    |
| 1   | PC1DR    | 0       | R/W | _                    |
| 0   | PC0DR    | 0       | R/W | _                    |

# Table 23.4 Read and Write Operations of Port C Data Register (PCDR)

• Bits 15 to 0 in PCDR

| PCIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | Pin state  | Data can be written to PCDR but data does not affect the pin state. |
|       | Other than general input  | Pin state  | Data can be written to PCDR but data does not affect the pin state. |
| 1     | General output            | PCDR value | Data written to PCDR is output on the pin.                          |
|       | Other than general output | PCDR value | Data can be written to PCDR but data does not affect the pin state. |

### 23.5.2 Port C Port Register (PCPR)

PCPR is a 16-bit read-only register that always stores port C pin states. This register cannot be directly written by the CPU. Bits PC15PR to PC0PR correspond to pins PC15/D15 to PC0/D0, respectively.

When PCPR is read, the pin state can be read.

| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|------------|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | PC15<br>PR | PC14<br>PR | PC13<br>PR | PC12<br>PR | PC11<br>PR | PC10<br>PR | PC9PR | PC8PR | PC7PR | PC6PR | PC5PR | PC4PR | PC3PR | PC2PR | PC1PR | PC0PR |
| Initial value: | PC15       | PC14       | PC13       | PC12       | PC11       | PC10       | PC9   | PC8   | PC7   | PC6   | PC5   | PC4   | PC3   | PC2   | PC1   | PC0   |
| ₽/M·           | R          | R          | R          | R          | R          | R          | R     | R     | R     | R     | R     | R     | R     | R     | R     | R     |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 15  | PC15PR   | Pin state        | R   | When these bits are read, the pin states can be read. |
| 14  | PC14PR   | Pin state        | R   | These bits cannot be modified.                        |
| 13  | PC13PR   | Pin state        | R   | -   |
| 12  | PC12PR   | Pin state        | R   | -   |
| 11  | PC11PR   | Pin state        | R   | -   |
| 10  | PC10PR   | Pin state        | R   | _   |
| 9   | PC9PR    | Pin state        | R   | -   |
| 8   | PC8PR    | Pin state        | R   | _   |
| 7   | PC7PR    | Pin state        | R   | -   |
| 6   | PC6PR    | Pin state        | R   | -   |
| 5   | PC5PR    | Pin state        | R   | -   |
| 4   | PC4PR    | Pin state        | R   | -   |
| 3   | PC3PR    | Pin state        | R   | -   |
| 2   | PC2PR    | Pin state        | R   | -   |
| 1   | PC1PR    | Pin state        | R   | <del>-</del>  |
| 0   | PC0PR    | Pin state        | R   | <del>-</del>  |

### 23.6 Port D

Port D is an input/output port with the 14 pins shown in figure 23.4.

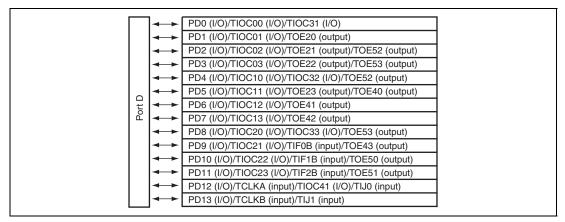


Figure 23.4 Port D

### 23.6.1 Port D Data Register (PDDR)

PDDR is a 16-bit readable/writable register that stores port D data. Bits PD13DR to PD0DR correspond to pins PD13/TCLKB/TIJ1 to PD0/TIOC00/TIOC31, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.5 summarizes the PDDR read and write operations.

PDDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | -  | -  | PD13<br>DR | PD12<br>DR | PD11<br>DR | PD10<br>DR | PD9DR | PD8DR | PD7DR | PD6DR | PD5DR | PD4DR | PD3DR | PD2DR | PD1DR | PD0DR |
| Initial value: | 0  | 0  | 0          | 0          | 0          | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W⋅           | R  | R  | R/M        | R/M        | R/M        | R/M        | R/M   | R/M   | R/M   | R/M   | R/M   | R/W   | R/M   | R/W   | R/M   | R/M   |

| Bit    | Bit Name | Initial<br>Value | R/W | Descriptions   |
|--------|----------|------------------|-----|--|
| 15, 14 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 13     | PD13DR   | 0                | R/W | Refer to table 23.5.   |
| 12     | PD12DR   | 0                | R/W | _  |
| 11     | PD11DR   | 0                | R/W | _  |
| 10     | PD10DR   | 0                | R/W | _  |
| 9      | PD9DR    | 0                | R/W | _  |
| 8      | PD8DR    | 0                | R/W | _  |
| 7      | PD7DR    | 0                | R/W | _  |
| 6      | PD6DR    | 0                | R/W | _  |
| 5      | PD5DR    | 0                | R/W | _  |
| 4      | PD4DR    | 0                | R/W | _  |
| 3      | PD3DR    | 0                | R/W | _  |
| 2      | PD2DR    | 0                | R/W | _  |
| 1      | PD1DR    | 0                | R/W | _  |
| 0      | PD0DR    | 0                | R/W | <del>-</del><br>   |

Table 23.5 Read and Write Operations of Port D Data Register (PDDR)

• Bits 13 to 0 in the PDDR register

| PDIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | Pin state  | Data can be written to PDDR but data does not affect the pin state. |
|       | Other than general input  | Pin state  | Data can be written to PDDR but data does not affect the pin state. |
| 1     | General output            | PDDR value | Data written to PDDR is output on the pin.                          |
|       | Other than general output | PDDR value | Data can be written to PDDR but data does not affect the pin state. |

# 23.6.2 Port D Port Register (PDPR)

PDPR is a 16-bit read-only register that always stores port D pin states. This register cannot be directly written by the CPU. The PD13PR to PB0PR bits correspond to the PD13/TCLKB/ TIJ1 to PD0/TIOC00/TIOC31 pins, respectively.

When PDPR is read, the pin state can be read.

| Bit:           | 15 | 14 | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | -  | -  | PD13<br>PR | PD12<br>PR | PD11<br>PR | PD10<br>PR | PD9PR | PD8PR | PD7PR | PD6PR | PD5PR | PD4PR | PD3PR | PD2PR | PD1PR | PD0PR |
| Initial value: | 0  | 0  | PD13       | PD12       | PD11       | PD10       | PD9   | PD8   | PD7   | PD6   | PD5   | PD4   | PD3   | PD2   | PD1   | PD0   |
| R/W·           | R  | R  | R          | R          | R          | R          | R     | R     | R     | R     | R     | R     | R     | R     | R     | R     |

| Bit    | Bit Name | Initial<br>Value | R/W | Descriptions   |
|--------|----------|------------------|-----|--|
| 15, 14 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 13     | PD13PR   | Pin state        | R   | When these bits are read, the pin states can be read.                |
| 12     | PD12PR   | Pin state        | R   | These bits cannot be modified.                                       |
| 11     | PD11PR   | Pin state        | R   | _  |
| 10     | PD10PR   | Pin state        | R   | _  |
| 9      | PD9PR    | Pin state        | R   | _  |
| 8      | PD8PR    | Pin state        | R   | _  |
| 7      | PD7PR    | Pin state        | R   | <del>-</del>   |
| 6      | PD6PR    | Pin state        | R   | _  |
| 5      | PD5PR    | Pin state        | R   | _  |
| 4      | PD4PR    | Pin state        | R   | _  |
| 3      | PD3PR    | Pin state        | R   | _  |
| 2      | PD2PR    | Pin state        | R   | <del>-</del>   |
| 1      | PD1PR    | Pin state        | R   | <del>-</del>   |
| 0      | PD0PR    | Pin state        | R   | _  |

### 23.6.3 Port D Inverting Register (PDIR)

PDIR is a 16-bit readable/writable register that enables or disables the inverting function for port D. Bits PD12IR to PD0IR correspond to pins PD12/TCLKA/TIOC41/TIJ0 to PD0/TIOC00/TIOC31, respectively. The PDIR setting is valid regardless of the port D pin functions.

Setting the specific bits of PDIR to 1 reverts the output values on their corresponding pins.

PDIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|----|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | -  | -  | -  | PD12IR | PD11IR | PD10IR | PD9IR | PD8IR | PD7IR | PD6IR | PD5IR | PD4IR | PD3IR | PD2IR | PD1IR | PD0IR |
| Initial value: | 0  | 0  | 0  | 0      | 0      | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W·           | R  | R  | R  | R/W    | R/W    | R/W    | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |

| 15 to 13  | Bit      | Bit Name | Initial<br>Value | R/W | Descriptions                              |
|---|----------|----------|------------------|-----|---|
| Should always be 0.   | 15 to 13 | _        | All 0            | R   | Reserved                                  |
| 11         PD11IR         0         R/W         when it is output on a corresponding pin.           10         PD10IR         0         R/W           9         PD9IR         0         R/W           8         PD8IR         0         R/W           7         PD7IR         0         R/W           6         PD6IR         0         R/W           5         PD5IR         0         R/W           4         PD4IR         0         R/W           3         PD3IR         0         R/W           1         PD1IR         0         R/W |          |          |                  |     | •   |
| 10  | 12       | PD12IR   | 0                | R/W |   |
| 9       PD10IR       0       R/W         8       PD8IR       0       R/W         7       PD7IR       0       R/W         6       PD6IR       0       R/W         5       PD5IR       0       R/W         4       PD4IR       0       R/W         3       PD3IR       0       R/W         2       PD2IR       0       R/W         1       PD1IR       0       R/W  | 11       | PD11IR   | 0                | R/W | ,   |
| 9  PD9IR  | 10       | PD10IR   | 0                | R/W |   |
| 7 PD7IR 0 R/W 6 PD6IR 0 R/W 5 PD5IR 0 R/W 4 PD4IR 0 R/W 3 PD3IR 0 R/W 2 PD2IR 0 R/W 1 PD1IR 0 R/W   | 9        | PD9IR    | 0                | R/W | -1: The set value is inverted when output |
| 6 PD6IR 0 R/W 5 PD5IR 0 R/W 4 PD4IR 0 R/W 3 PD3IR 0 R/W 2 PD2IR 0 R/W 1 PD1IR 0 R/W   | 8        | PD8IR    | 0                | R/W | _   |
| 5     PD5IR     0     R/W       4     PD4IR     0     R/W       3     PD3IR     0     R/W       2     PD2IR     0     R/W       1     PD1IR     0     R/W   | 7        | PD7IR    | 0                | R/W | _   |
| 4 PD4IR 0 R/W 3 PD3IR 0 R/W 2 PD2IR 0 R/W 1 PD1IR 0 R/W   | 6        | PD6IR    | 0                | R/W | _   |
| 3 PD3IR 0 R/W 2 PD2IR 0 R/W 1 PD1IR 0 R/W   | 5        | PD5IR    | 0                | R/W | _   |
| 2 PD2IR 0 R/W 1 PD1IR 0 R/W   | 4        | PD4IR    | 0                | R/W |   |
| 1 PD1IR 0 R/W   | 3        | PD3IR    | 0                | R/W |   |
|   | 2        | PD2IR    | 0                | R/W |   |
| 0 PD0IR 0 R/W   | 1        | PD1IR    | 0                | R/W |   |
|   | 0        | PD0IR    | 0                | R/W |   |

### 23.7 Port E

Port E is an input/output port with the 14 pins shown in figure 23.5.

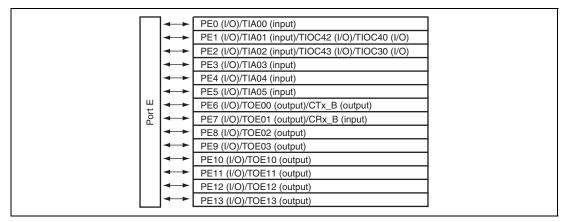


Figure 23.5 Port E

### 23.7.1 Port E Data Register (PEDR)

PEDR is a 16-bit readable/writable register that stores port E data. Bits PE13DR to PE0DR correspond to pins PE13/TOE13 to PE0/TIA00, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.6 summarizes the PEDR read and write operations.

PEDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | -  | -  | PE13<br>DR | PE12<br>DR | PE11<br>DR | PE10<br>DR | PE9DR | PE8DR | PE7DR | PE6DR | PE5DR | PE4DR | PE3DR | PE2DR | PE1DR | PE0DR |
| Initial value: | 0  | 0  | 0          | 0          | 0          | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/M⋅           | R  | R  | R/M        | R/M        | R/W        | R/M        | R/W   | R/M   | R/M   | R/M   | R/M   | R/M   | R/W   | R/M   | R/M   | R/M   |

| Bit    | Bit Name | Initial<br>Value | R/W | Descriptions   |
|--------|----------|------------------|-----|--|
| 15, 14 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 13     | PE13DR   | 0                | R/W | Refer to table 23.6.   |
| 12     | PE12DR   | 0                | R/W | <del>-</del>   |
| 11     | PE11DR   | 0                | R/W | _  |
| 10     | PE10DR   | 0                | R/W | _  |
| 9      | PE9DR    | 0                | R/W | _  |
| 8      | PE8DR    | 0                | R/W | _  |
| 7      | PE7DR    | 0                | R/W | _  |
| 6      | PE6DR    | 0                | R/W | _  |
| 5      | PE5DR    | 0                | R/W | _  |
| 4      | PE4DR    | 0                | R/W | _  |
| 3      | PE3DR    | 0                | R/W | _  |
| 2      | PE2DR    | 0                | R/W | _  |
| 1      | PE1DR    | 0                | R/W | _  |
| 0      | PE0DR    | 0                | R/W |  |

# Table 23.6 Read and Write Operations of Port E Data Register (PEDR)

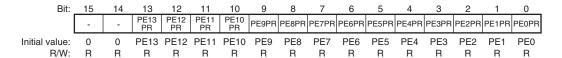
• Bits 13 to 0 in the PEDR register

| PEIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | Pin state  | Data can be written to PEDR but data does not affect the pin state. |
|       | Other than general input  | Pin state  | Data can be written to PEDR but data does not affect the pin state. |
| 1     | General output            | PEDR value | Data written to PEDR is output on the pin.                          |
|       | Other than general output | PEDR value | Data can be written to PEDR but data does not affect the pin state. |

### 23.7.2 Port E Port Register (PEPR)

PEPR is a 16-bit read-only register that always stores port E pin states. This register cannot be directly written by the CPU. The PE13PR to PE0PR bits correspond to the PE13/TOE13 to PE0/TIA00 pins, respectively.

When PEPR is read, the pin state can be read.



| Bit    | Bit Name | Initial<br>Value | R/W | Descriptions   |
|--------|----------|------------------|-----|--|
| 15, 14 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 13     | PE13PR   | Pin state        | R   | When these bits are read, the pin states can be read.                |
| 12     | PE12PR   | Pin state        | R   | These bits cannot be modified.                                       |
| 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   | _  |

#### **Port E Inverting Register (PEIR)** 23.7.3

PEIR is a 16-bit readable/writable register that enables or disables the inverting function for port E. Bits PE13IR to PE6IR, PE2IR, and PE1IR correspond to pins PE13/TOE13 to PE6/TOE00/CTx\_B, PE2/TIA02/TIOC43/TIOC30, PE1/TIA01/TIOC42/TIOC40, respectively. The PEIR setting is valid regardless of the port E pin functions.

Setting the specific bits of PEIR to 1 reverts the output values on their corresponding pins.

PEIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5 | 4 | 3 | 2     | 1     | 0 |
|----------------|----|----|--------|--------|--------|--------|-------|-------|-------|-------|---|---|---|-------|-------|---|
|                | -  | -  | PE13IR | PE12IR | PE11IR | PE10IR | PE9IR | PE8IR | PE7IR | PE6IR | - | - | - | PE2IR | PE1IR | - |
| Initial value: | 0  | 0  | 0      | 0      | 0      | 0      | 0     | 0     | 0     | 0     | 0 | 0 | 0 | 0     | 0     | 0 |
| R/W·           | R  | R  | R/W    | R/W    | R/W    | R/W    | R/W   | R/W   | R/W   | R/W   | R | R | R | R/W   | R/W   | R |

| Bit    | Bit Name | Initial<br>Value | R/W | Descriptions   |
|--------|----------|------------------|-----|--|
| 15, 14 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 13     | PE13IR   | 0                | R/W | These bits set whether or not the set value is inverted              |
| 12     | PE12IR   | 0                | R/W | when it is output on a corresponding pin.                            |
| 11     | PE11IR   | 0                | R/W | <sup>-</sup> 0: The set value is not inverted when output            |
| 10     | PE10IR   | 0                | R/W | 1: The set value is inverted when output                             |
| 9      | PE9IR    | 0                | R/W | _  |
| 8      | PE8IR    | 0                | R/W | _  |
| 7      | PE7IR    | 0                | R/W | _  |
| 6      | PE6IR    | 0                | R/W | <del>-</del>   |
| 5 to 3 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 2      | PE2IR    | 0                | R/W | These bits set whether or not the set value is inverted              |
| 1      | PE1IR    | 0                | R/W | when it is output on a corresponding pin.                            |
|        |          |                  |     | 0: The set value is not inverted when output                         |
| -      |          |                  |     | 1: The set value is inverted when output                             |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 0   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0. |

### 23.7.4 Port E Driving Ability Setting Register (PEDSR)

PEDSR is a 16-bit readable/writable register that selects the driving ability for port E. Bits PE10DSR to PE6DSR correspond to pins PE10/TOE10 to PE6/TOE00/CTx\_B, respectively. The PEDSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PEDSR to 1 increases the driving ability of their corresponding pins.

PEDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10          | 9          | 8          | 7          | 6          | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|----|----|----|----|----|-------------|------------|------------|------------|------------|---|---|---|---|---|---|
|                | -  | -  | -  | -  | -  | PE10<br>DSR | PE9<br>DSR | PE8<br>DSR | PE7<br>DSR | PE6<br>DSR | - | - | - | - | - | - |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0           | 0          | 0          | 0          | 0          | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W:           | R  | R  | R  | R  | R  | R/W         | R/W        | R/W        | R/W        | R/W        | R | R | R | R | R | R |

| Bit      | Bit Name | Initial<br>Value | R/W | Descriptions  |
|----------|----------|------------------|-----|---|
| 15 to 11 | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.                              |
| 10       | PE10DSR  | 0                | R/W | When these bits are set to 1, the driving ability of the  |
| 9        | PE9DSR   | 0                | R/W | corresponding pin is higher than normal.  |
| 8        | PE8DSR   | 0                | R/W | O: Normal driving ability (slow slew rate)  |
| 7        | PE7DSR   | 0                | R/W | 1: High driving ability (fast slew rate)  |
| 6        | PE6DSR   | 0                | R/W | Note: For the characteristics of the driving ability, refer to section 32.3.14, Output Slew Rate. |
| 5 to 0   | _        | All 0            | R   | Reserved  |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.                              |

#### 23.8 Port F

Port F is an input/output port with the 16 pins shown in figure 23.6.

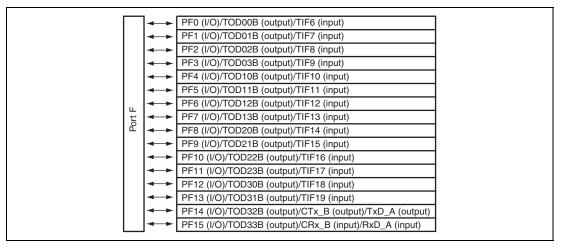


Figure 23.6 Port F

#### 23.8.1 Port F Data Register (PFDR)

PFDR is a 16-bit readable/writable register that stores port F data. Bits PF15DR to PF0DR correspond to pins PF15/TOD33B/CRx B/RxD A to PF0/TOD00B/TIF6, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.7 summarizes the PFDR read and write operations.

PFDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|------------|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | PF15<br>DR | PF14<br>DR | PF13<br>DR | PF12<br>DR | PF11<br>DR | PF10<br>DR | PF9DR | PF8DR | PF7DR | PF6DR | PF5DR | PF4DR | PF3DR | PF2DR | PF1DR | PF0DR |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| ₽/M/·          | R/M        | D/M        | D/M        | D/\/       | D/\//      | D/W        | D/\/  | D/W   | R/W   | P/M   | D/M   | P/M   | R/W   | P/M   | D/M   | D/W   |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions         |
|-----|----------|------------------|-----|----------------------|
|     |          |                  |     |                      |
| 15  | PF15DR   | 0                | R/W | Refer to table 23.7. |
| 14  | PF14DR   | 0                | R/W | _                    |
| 13  | PF13DR   | 0                | R/W | <del>-</del>         |
| 12  | PF12DR   | 0                | R/W | _                    |
| 11  | PF11DR   | 0                | R/W | <del>-</del>         |
| 10  | PF10DR   | 0                | R/W | <del>-</del>         |
| 9   | PF9DR    | 0                | R/W | <del>-</del>         |
| 8   | PF8DR    | 0                | R/W | _                    |
| 7   | PF7DR    | 0                | R/W | <del>-</del>         |
| 6   | PF6DR    | 0                | R/W | <del>-</del>         |
| 5   | PF5DR    | 0                | R/W | _                    |
| 4   | PF4DR    | 0                | R/W | <del>-</del>         |
| 3   | PF3DR    | 0                | R/W | _                    |
| 2   | PF2DR    | 0                | R/W | _                    |
| 1   | PF1DR    | 0                | R/W | _                    |
| 0   | PF0DR    | 0                | R/W | _                    |

# Table 23.7 Read and Write Operations of Port F Data Register (PFDR)

• Bits 15 to 0 in the PFDR register

| PFIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | Pin state  | Data can be written to PFDR but data does not affect the pin state. |
|       | Other than general input  | Pin state  | Data can be written to PFDR but data does not affect the pin state. |
| 1     | General output            | PFDR value | Data written to PFDR is output on the pin.                          |
|       | Other than general output | PFDR value | Data can be written to PFDR but data does not affect the pin state. |

# 23.8.2 Port F Port Register (PFPR)

PFPR is a 16-bit read-only register that always stores port F pin states. This register cannot be directly written by the CPU. Bits PF15PR to PF0PR correspond to pins PF15/TOD33B/CRx\_B/RxD\_A to PF0/TOD00B/TIF6, respectively.

When PFPR is read, the pin state can be read.

| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|------------|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | PF15<br>PR | PF14<br>PR | PF13<br>PR | PF12<br>PR | PF11<br>PR | PF10<br>PR | PF9PR | PF8PR | PF7PR | PF6PR | PF5PR | PF4PR | PF3PR | PF2PR | PF1PR | PF0PR |
| Initial value: | PF15       | PF14       | PF13       | PF12       | PF11       | PF10       | PF9   | PF8   | PF7   | PF6   | PF5   | PF4   | PF3   | PF2   | PF1   | PF0   |
| R/W:           | R          | R          | R          | R          | R          | R          | R     | R     | R     | R     | R     | R     | R     | R     | R     | R     |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 15  | PF15PR   | Pin state        | R   | When these bits are read, the pin states can be read. |
| 14  | PF14PR   | Pin state        | R   | These bits cannot be modified.                        |
| 13  | PF13PR   | Pin state        | R   | -   |
| 12  | PF12PR   | Pin state        | R   | <del>-</del>  |
| 11  | PF11PR   | Pin state        | R   | <del>-</del>  |
| 10  | PF10PR   | Pin state        | R   | -   |
| 9   | PF9PR    | Pin state        | R   | -   |
| 8   | PF8PR    | Pin state        | R   | <del>-</del>  |
| 7   | PF7PR    | Pin state        | R   | _   |
| 6   | PF6PR    | Pin state        | R   | -   |
| 5   | PF5PR    | Pin state        | R   | <del>-</del>  |
| 4   | PF4PR    | Pin state        | R   | -   |
| 3   | PF3PR    | Pin state        | R   | -   |
| 2   | PF2PR    | Pin state        | R   | -   |
| 1   | PF1PR    | Pin state        | R   | <del>-</del>  |
| 0   | PF0PR    | Pin state        | R   | -   |

#### **Port F Inverting Register (PFIR)** 23.8.3

PFIR is a 16-bit readable/writable register that enables or disables the inverting function for port F. Bits PF15IR to PF0IR correspond to pins PF15/TOD33B/CRx\_B/RxD\_A to PF0/TOD00B/TIF6, respectively. The PFIR setting is valid regardless of the port F pin functions.

Setting the specific bits of PFIR to 1 reverts the output values on their corresponding pins.

PFIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | PF15IR | PF14IR | PF13IR | PF12IR | PF11IR | PF10IR | PF9IR | PF8IR | PF7IR | PF6IR | PF5IR | PF4IR | PF3IR | PF2IR | PF1IR | PF0IR |
| Initial value: | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W·           | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 15  | PF15IR   | 0                | R/W | These bits set whether or not the set value is inverted |
| 14  | PF14IR   | 0                | R/W | when it is output on a corresponding pin.               |
| 13  | PF13IR   | 0                | R/W | 0: The set value is not inverted when output            |
| 12  | PF12IR   | 0                | R/W | -1: The set value is inverted when output               |
| 11  | PF11IR   | 0                | R/W | _   |
| 10  | PF10IR   | 0                | R/W | _   |
| 9   | PF9IR    | 0                | R/W | _   |
| 8   | PF8IR    | 0                | R/W | _   |
| 7   | PF7IR    | 0                | R/W | _   |
| 6   | PF6IR    | 0                | R/W | _   |
| 5   | PF5IR    | 0                | R/W | _   |
| 4   | PF4IR    | 0                | R/W | <del>-</del>  |
| 3   | PF3IR    | 0                | R/W | _   |
| 2   | PF2IR    | 0                | R/W | _   |
| 1   | PF1IR    | 0                | R/W |   |
| 0   | PF0IR    | 0                | R/W | _   |

### 23.8.4 Port F Driving Ability Setting Register (PFDSR)

PFDSR is a 16-bit readable/writable register that selects the driving ability for port F. Bit PF14DSR corresponds to pin PF14/TOD32B/CTx\_B/TxD\_A. The PFDSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PFDSR to 1 increases the driving ability of their corresponding pins.

PFDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14          | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|----|-------------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|                | -  | PF14<br>DSR | -  | -  | -  | -  | - | - | - | - | - | - | - | - | - | - |
| Initial value: | 0  | 0           | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W:           | R  | R/W         | R  | R  | R  | R  | R | R | R | R | R | R | R | R | R | R |

| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions  |
|---------|----------|------------------|-----|---|
| 15      | _        | 0                | R   | Reserved  |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0.                                 |
| 14      | PF14DSR  | 0                | R/W | When this bit is set to 1, the driving ability of the corresponding pin is higher than normal.    |
|         |          |                  |     | 0: Normal driving ability (slow slew rate)  |
|         |          |                  |     | 1: High driving ability (fast slew rate)  |
|         |          |                  |     | Note: For the characteristics of the driving ability, refer to section 32.3.14, Output Slew Rate. |
| 13 to 0 | _        | All 0            | R   | Reserved  |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0.                              |

# 23.8.5 Port F Pin State Setting Register (PFPSR)

PFPSR is a 16-bit readable/writable register that enables or disables the pull-down state of port F. Bit PF15PSR corresponds to pins PF15/TOD33B/CRx\_B/RxD\_A. By setting this bit to 1, the signal on its corresponding pin pulled down though the pull-down resistor.

PFPSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|-------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|                | PF15<br>PSR | -  | -  | -  | -  | -  | - | - | - | - | 1 | - | - | - | - | - |
| Initial value: | 0           | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W·           | R/W         | R  | R  | R  | R  | R  | R | R | R | R | R | R | R | R | R | R |

|         |          | Initial |     |  |
|---------|----------|---------|-----|--|
| Bit     | Bit Name | Value   | R/W | Descriptions   |
| 15      | PF15PSR  | 0       | R/W | Selects whether or not to enable the pull-down resistor of a corresponding pin is enabled. |
|         |          |         |     | 0: Pull-down resistor disabled   |
|         |          |         |     | 1: Pull-down resistor enabled  |
| 14 to 0 | _        | All 0   | R   | Reserved   |
|         |          |         |     | These bits are always read as 0. The write value should always be 0.                       |

### 23.9 Port G

Port G is an input/output port with the 16 pins shown in figure 23.7.

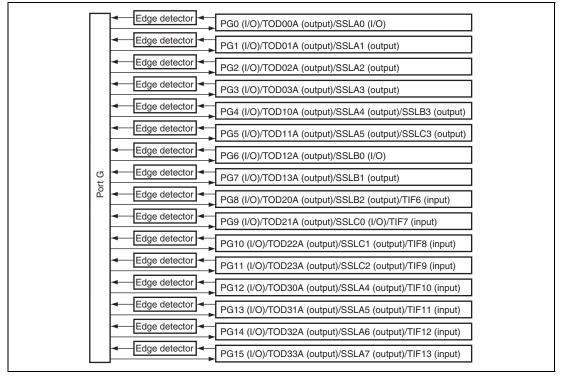


Figure 23.7 Port G

# 23.9.1 Port G Data Register (PGDR)

PGDR is a 16-bit readable/writable register that stores port G data. Bits PG15DR to PG0DR correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the bit value can be read. Each bit of PGDR is set to 1 when an edge specified by the port G edge selecting register (PGER) is detected on the corresponding pin. Only 0 can be written after the bit is read as 1. Writing 0

without having read the bit as 1 or writing 1 is ignored. Writing 0 to the bits does not affect the pin state. Table 23.8 summarizes the PGDR read and write operations.

PGDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|------------|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | PG15<br>DR | PG14<br>DR | PG13<br>DR | PG12<br>DR | PG11<br>DR | PG10<br>DR | PG9DR | PG8DR | PG7DR | PG6DR | PG5DR | PG4DR | PG3DR | PG2DR | PG1DR | PG0DR |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |

|     |          | Initial |     |                      |
|-----|----------|---------|-----|----------------------|
| Bit | Bit Name | Value   | R/W | Descriptions         |
| 15  | PG15DR   | 0       | R/W | Refer to table 23.8. |
| 14  | PG14DR   | 0       | R/W | _                    |
| 13  | PG13DR   | 0       | R/W | _                    |
| 12  | PG12DR   | 0       | R/W | _                    |
| 11  | PG11DR   | 0       | R/W | <del>-</del>         |
| 10  | PG10DR   | 0       | R/W | <del>-</del>         |
| 9   | PG9DR    | 0       | R/W | <del>-</del>         |
| 8   | PG8DR    | 0       | R/W | <del>-</del>         |
| 7   | PG7DR    | 0       | R/W | <del>-</del>         |
| 6   | PG6DR    | 0       | R/W | _                    |
| 5   | PG5DR    | 0       | R/W | <del>-</del>         |
| 4   | PG4DR    | 0       | R/W | <del>-</del>         |
| 3   | PG3DR    | 0       | R/W | <del>-</del>         |
| 2   | PG2DR    | 0       | R/W | _                    |
| 1   | PG1DR    | 0       | R/W | _                    |
| 0   | PG0DR    | 0       | R/W | _                    |

# Table 23.8 Read and Write Operations of Port G Data Register (PGDR)

• Bits 15 to 0 in the PGDR register

| PGIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | PGDR value | Writing 1 is ignored. Only 0 can be written after this bit is read as 1. Writing 0 to this bit does not affect the pin state. |
|       |                           |            | Each bit can be set to 1 when an edge specified by PGER is input on the corresponding pin.                                    |
|       | Other than general input  | PGDR value | Writing 1 is ignored. Only 0 can be written after this bit is read as 1. Writing 0 to this bit does not affect the pin state. |
|       |                           |            | Each bit can be set to 1 when an edge specified by PGER is input/output on the corresponding pin.                             |
| 1     | General output            | PGDR value | Data written to PGDR is output on the pin.  |
|       | Other than general output | PGDR value | Data can be written to PGDR but data does not affect the pin state.   |

#### 23.9.2 Port G Port Register (PGPR)

PGPR is a 16-bit read-only register that always stores port G pin states. This register cannot be directly written by the CPU. Bits PG15PR to PG0PR correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0, respectively.

When PGPR is read, the pin state can be read.

| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|------------|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | PG15<br>PR | PG14<br>PR | PG13<br>PR | PG12<br>PR | PG11<br>PR | PG10<br>PR | PG9PR | PG8PR | PG7PR | PG6PR | PG5PR | PG4PR | PG3PR | PG2PR | PG1PR | PG0PR |
| Initial value: | PG15       | PG14       | PG13       | PG12       | PG11       | PG10       | PG9   | PG8   | PG7   | PG6   | PG5   | PG4   | PG3   | PG2   | PG1   | PG0   |
| R/W:           | R          | R          | R          | R          | R          | R          | R     | R     | R     | R     | R     | R     | R     | R     | R     | R     |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 15  | PG15PR   | Pin state        | R   | When these bits are read, the pin states can be read. |
| 14  | PG14PR   | Pin state        | R   | These bits cannot be modified.                        |
| 13  | PG13PR   | Pin state        | R   | -   |
| 12  | PG12PR   | Pin state        | R   | _   |
| 11  | PG11PR   | Pin state        | R   | _   |
| 10  | PG10PR   | Pin state        | R   | _   |
| 9   | PG9PR    | Pin state        | R   | _   |
| 8   | PG8PR    | Pin state        | R   | _   |
| 7   | PG7PR    | Pin state        | R   |   |
| 6   | PG6PR    | Pin state        | R   | _   |
| 5   | PG5PR    | Pin state        | R   | -   |
| 4   | PG4PR    | Pin state        | R   | _   |
| 3   | PG3PR    | Pin state        | R   | _   |
| 2   | PG2PR    | Pin state        | R   | -   |
| 1   | PG1PR    | Pin state        | R   | _   |
| 0   | PG0PR    | Pin state        | R   | -   |

# 23.9.3 Port G Inverting Register (PGIR)

PGIR is a 16-bit readable/writable register that enables or disables the inverting function for port G. Bits PG15IR to PG0IR correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0, respectively. The PGIR setting is valid regardless of the port G pin functions.

Setting a bit of PGIR to 1 reverts the output value on its corresponding pin.

The PGIR register is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15         | 14         | 13         | 12         | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|------------|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | PG15<br>IR | PG14<br>IR | PG13<br>IR | PG12<br>IR | PG11<br>IR | PG10<br>IR | PG9IR | PG8IR | PG7IR | PG6IR | PG5IR | PG4IR | PG3IR | PG2IR | PG1IR | PG0IR |
| Initial value: | 0          | 0          | 0          | 0          | 0          | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

| <b></b> | DI: N    | Initial | - n., | <b>5</b>  |
|---------|----------|---------|-------|---|
| Bit     | Bit Name | Value   | R/W   | Descriptions  |
| 15      | PG15IR   | 0       | R/W   | These bits set whether or not the set value is inverted |
| 14      | PG14IR   | 0       | R/W   | when it is output on a corresponding pin.               |
| 13      | PG13IR   | 0       | R/W   | O: The set value is not inverted when output            |
| 12      | PG12IR   | 0       | R/W   | −1: The set value is inverted when output               |
| 11      | PG11IR   | 0       | R/W   | <del>-</del>  |
| 10      | PG10IR   | 0       | R/W   | <del>-</del>  |
| 9       | PG9IR    | 0       | R/W   | _   |
| 8       | PG8IR    | 0       | R/W   | <del>-</del>  |
| 7       | PG7IR    | 0       | R/W   | <del>-</del>  |
| 6       | PG6IR    | 0       | R/W   | <del>-</del>  |
| 5       | PG5IR    | 0       | R/W   | <del>-</del>  |
| 4       | PG4IR    | 0       | R/W   | <del>-</del>  |
| 3       | PG3IR    | 0       | R/W   | _   |
| 2       | PG2IR    | 0       | R/W   | _   |
| 1       | PG1IR    | 0       | R/W   | _   |
| 0       | PG0IR    | 0       | R/W   | _   |
|         |          |         |       |   |

### 23.9.4 Port G Driving Ability Setting Register (PGDSR)

PGDSR is a 16-bit readable/writable register that selects the driving ability for port G. Bits PG15DSR to PG0DSR correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0, respectively. The PGDSR setting is valid regardless of the selected pin functions.

When a bit in PGDSR is set to 1, the high driving ability is set in the corresponding pin.

PGDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. PGDSR is not initialized in sleep mode.

| Bit:           | 15          | 14          | 13          | 12          | 11          | 10          | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|                | PG15<br>DSR | PG14<br>DSR | PG13<br>DSR | PG12<br>DSR | PG11<br>DSR | PG10<br>DSR | PG9<br>DSR | PG8<br>DSR | PG7<br>DSR | PG6<br>DSR | PG5<br>DSR | PG4<br>DSR | PG3<br>DSR | PG2<br>DSR | PG1<br>DSR | PG0<br>DSR |
| Initial value: | 0           | 0           | 0           | 0           | 0           | 0           | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 15  | PG15DSR  | 0                | R/W | When these bits are set to 1, the driving ability of the  |
| 14  | PG14DSR  | 0                | R/W | corresponding pin is higher than normal.  |
| 13  | PG13DSR  | 0                | R/W | Normal driving ability (slow slew rate)   |
| 12  | PG12DSR  | 0                | R/W | -1: High driving ability (fast slew rate)   |
| 11  | PG11DSR  | 0                | R/W | <ul> <li>Note: For the characteristics of the driving ability, refer<br/>to section 32.3.14, Output Slew Rate.</li> </ul> |
| 10  | PG10DSR  | 0                | R/W | <del>-</del>  |
| 9   | PG9DSR   | 0                | R/W | <del>-</del>  |
| 8   | PG8DSR   | 0                | R/W | _   |
| 7   | PG7DSR   | 0                | R/W | _   |
| 6   | PG6DSR   | 0                | R/W | _   |
| 5   | PG5DSR   | 0                | R/W | _   |
| 4   | PG4DSR   | 0                | R/W | _   |
| 3   | PG3DSR   | 0                | R/W | _   |
| 2   | PG2DSR   | 0                | R/W |   |
| 1   | PG1DSR   | 0                | R/W |   |
| 0   | PG0DSR   | 0                | R/W | _   |

### 23.9.5 Port G Edge Selecting Register (PGER)

PGER is a 16-bit readable/writable register that selects the type of an edge to be detected by port G. The PGHES[1:0] bits correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG8/TOD20A/SSLB2/TIF6. The PGLES[1:0] bits correspond to pins PG7/TOD13A/SSLB1 to PG0/TOD00A/SSLA0 pins.

When a pin functions as a general input, each bit in PGDR functions as a status flag indicating whether or not the specified edge input is detected.

PGER is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

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

| Bit      | Bit Name   | Initial<br>Value | R/W | Descriptions   |
|----------|------------|------------------|-----|--|
| 15 to 10 | _          | All 0            | R   | Reserved   |
|          |            |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 9, 8     | PGHES[1:0] | 00               | R/W | These bits set the input edge to be detected on the corresponding pins PG15/TOD33A/SSLA7/TIF13 to PG8/TOD20A/SSLB2/TIF6. |
|          |            |                  |     | 00*: No edge detected  |
|          |            |                  |     | 01: Rising edge detected   |
|          |            |                  |     | 10: Falling edge detected  |
|          |            |                  |     | 11: Both rising and falling edges detected   |
| 7 to 2   | _          | All 0            | R   | Reserved   |
|          |            |                  |     | These bits are always read as 0. The write value should always be 0.   |

|      |            | Initial |     |  |
|------|------------|---------|-----|--|
| Bit  | Bit Name   | Value   | R/W | Descriptions   |
| 1, 0 | PGLES[1:0] | 00      | R/W | These bits set the input edge to be detected on the corresponding pins PG7/TOD13A/SSLB1 to |
|      |            |         |     | PG0/TOD00A/SSLA0.  |
|      |            |         |     | 00*: No edge detected  |
|      |            |         |     | 01: Rising edge detected   |
|      |            |         |     | 10: Falling edge detected  |
|      |            |         |     | 11: Both rising and falling edges detected   |

### 23.10 Port H

Port H is an input/output port with the 6 pins shown in figure 23.8.

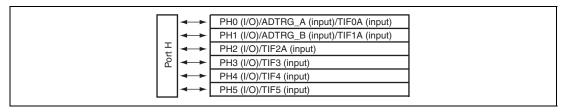


Figure 23.8 Port H

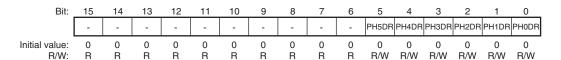
### 23.10.1 Port H Data Register (PHDR)

PHDR is a 16-bit readable/writable register that stores port H data. Bits PH5DR to PH0DR correspond to pins PH5/TIF5 to PH0/ADTRG\_A/TIF0A, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.9 summarizes the PHDR read and write operations.

PHDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.



| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions   |
|---------|----------|------------------|-----|--|
| 15 to 6 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 5       | PH5DR    | 0                | R/W | Refer to table 23.9.   |
| 4       | PH4DR    | 0                | R/W | <del>-</del>   |
| 3       | PH3DR    | 0                | R/W | _  |
| 2       | PH2DR    | 0                | R/W | _  |
| 1       | PH1DR    | 0                | R/W | _  |
| 0       | PH0DR    | 0                | R/W | _  |

# Table 23.9 Read and Write Operations of Port H Data Register (PHDR)

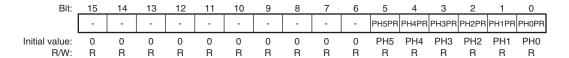
Bits 5 to 0 in the PHDR register

| PHIOR | Pin Function              | Read       | Write   |  |  |  |
|-------|---------------------------|------------|---|--|--|--|
| 0     | General input             | Pin state  | Data can be written to PHDR but data does not affect the pin state. |  |  |  |
|       | Other than general input  | Pin state  | Data can be written to PHDR but data does not affect the pin state. |  |  |  |
| 1     | General output            | PHDR value | Data written to PHDR is output on the pin.                          |  |  |  |
|       | Other than general output | PHDR value | Data can be written to PHDR but data does not affect the pin state. |  |  |  |

### 23.10.2 Port H Port Register (PHPR)

PHPR is a 16-bit read-only register that always stores port H pin states. This register cannot be directly written by the CPU. Bits PH5PR to PH0PR correspond to pins PH5/TIF5 to PH0/ADTRG\_A/TIF0A, respectively.

When PHPR is read, the pin state can be read.



| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions   |
|---------|----------|------------------|-----|--|
| 15 to 6 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 5       | PH5PR    | Pin state        | R   | When these bits are read, the pin states can be read.                |
| 4       | PH4PR    | Pin state        | R   | These bits cannot be modified.                                       |
| 3       | PH3PR    | Pin state        | R   | <del>-</del>   |
| 2       | PH2PR    | Pin state        | R   | <del>-</del>   |
| 1       | PH1PR    | Pin state        | R   | <del>-</del>   |
| 0       | PH0PR    | Pin state        | R   | _  |

### 23.11 Port J

Port J is an input/output port with the 10 pins shown in figure 23.9.

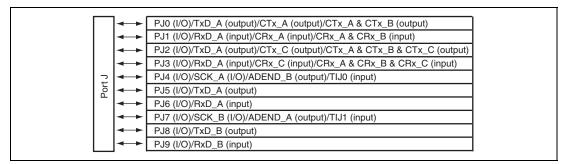


Figure 23.9 Port J

### 23.11.1 Port J Data Register (PJDR)

PJDR is a 16-bit readable/writable register that stores port J data. Bits PJ9DR to PJ0DR correspond to pins PJ9/RxD B to PJ0/TxD A/CTx A/CTx A&CTx B, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.10 summarizes the PJDR read and write operations.

PJDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15     | 14     | 13     | 12     | 11     | 10     | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|----------------|--------|--------|--------|--------|--------|--------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                | -      | -      | -      | -      | -      | -      | PJ9DR    | PJ8DR    | PJ7DR    | PJ6DR    | PJ5DR    | PJ4DR    | PJ3DR    | PJ2DR    | PJ1DR    | PJ0DR    |
| Initial value: | 0<br>B | 0<br>B | 0<br>B | 0<br>B | 0<br>B | 0<br>B | 0<br>B/W | 0<br>B/W | 0<br>B/W | 0<br>B/W | 0<br>R/W | 0<br>B/W | 0<br>R/W | 0<br>B/W | 0<br>R/W | 0<br>B/W |

| Bit      | Bit Name | Initial<br>Value | R/W | Descriptions   |
|----------|----------|------------------|-----|--|
| 15 to 10 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 9        | PJ9DR    | 0                | R/W | Refer to table 23.10.  |
| 8        | PJ8DR    | 0                | R/W | <del>-</del>   |
| 7        | PJ7DR    | 0                | R/W | _  |
| 6        | PJ6DR    | 0                | R/W | _  |
| 5        | PJ5DR    | 0                | R/W | _  |
| 4        | PJ4DR    | 0                | R/W | _  |
| 3        | PJ3DR    | 0                | R/W | _  |
| 2        | PJ2DR    | 0                | R/W | _  |
| 1        | PJ1DR    | 0                | R/W | _  |
| 0        | PJ0DR    | 0                | R/W | _  |

Table 23.10 Read and Write Operations of Port J Data Register (PJDR)

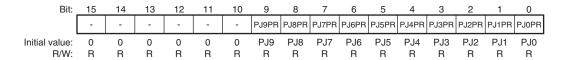
• Bits 9 to 0 in the PJDR register

| PJIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | Pin state  | Data can be written to PJDR but data does not affect the pin state. |
|       | Other than general input  | Pin state  | Data can be written to PJDR but data does not affect the pin state. |
| 1     | General output            | PJDR value | Data written to PJDR is output on the pin.                          |
|       | Other than general output | PJDR value | Data can be written to PJDR but data does not affect the pin state. |

# 23.11.2 Port J Port Register (PJPR)

PJPR is a 16-bit read-only register that always stores port J pin states. This register cannot be directly written by the CPU. Bits PJ9PR to PJ0PR correspond to pins PJ9/RxD\_B to PJ0/TxD\_A/CTx\_A/CTx\_A&CTx\_B, respectively.

When PJPR is read, the pin state can be read.



| Bit      | Bit Name | Initial<br>Value | R/W | Descriptions   |
|----------|----------|------------------|-----|--|
| 15 to 10 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 9        | PJ9PR    | Pin state        | R   | When these bits are read, the pin states can be read.                |
| 8        | PJ8PR    | Pin state        | R   | These bits cannot be modified.                                       |
| 7        | PJ7PR    | Pin state        | R   |  |
| 6        | PJ6PR    | Pin state        | R   | _  |
| 5        | PJ5PR    | Pin state        | R   | _  |
| 4        | PJ4PR    | Pin state        | R   |  |
| 3        | PJ3PR    | Pin state        | R   |  |
| 2        | PJ2PR    | Pin state        | R   |  |
| 1        | PJ1PR    | Pin state        | R   | _  |
| 0        | PJ0PR    | Pin state        | R   | _  |

SH72546R Section 23 I/O Ports

# 23.11.3 Port J Inverting Register (PJIR)

PJIR is a 16-bit readable/writable register that enables or disables the inverting function for port J. Bits PJ8IR, PJ7IR, PJ5IR, PJ4IR, PJ2IR, and PJ0IR correspond to pins PJ8/TxD B, PJ7/SCK\_B/ADEND\_A/TIJ1, PJ5/TxD\_A, PJ4/SCK\_A/ADEND\_B/TIJ0, PJ2/TxD\_A/CTx\_C/CTx\_A&CTx\_B&CTx\_C, and PJ0/TxD\_A/CTx\_A/CTx\_A&CTx\_B, respectively. The PJIR setting is valid regardless of the port J pin functions.

Setting the specific bits of PJIR to 1 inverts the output value on their corresponding pins.

PJIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7     | 6 | 5     | 4     | 3 | 2     | 1 | 0     |
|----------------|----|----|----|----|----|----|---|-------|-------|---|-------|-------|---|-------|---|-------|
|                | -  | -  | -  | -  | -  | -  | - | PJ8IR | PJ7IR | - | PJ5IR | PJ4IR | - | PJ2IR | - | PJ0IR |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0     | 0     | 0 | 0     | 0     | 0 | 0     | 0 | 0     |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R/W   | R/W   | R | R/W   | R/W   | R | R/W   | R | R/W   |

| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions   |
|---------|----------|------------------|-----|--|
| 15 to 9 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 8       | PJ8IR    | 0                | R/W | These bits set whether or not the set value is inverted              |
| 7       | PJ7IR    | 0                | R/W | when it is output on a corresponding pin.                            |
|         |          |                  |     | 0: The set value is not inverted when output                         |
|         |          |                  |     | 1: The set value is inverted when output                             |
| 6       | _        | 0                | R   | Reserved   |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0.    |
| 5       | PJ5IR    | 0                | R/W | These bits set whether or not the set value is inverted              |
| 4       | PJ4IR    | 0                | R/W | when it is output on a corresponding pin.                            |
|         |          |                  |     | 0: The set value is not inverted when output                         |
|         |          |                  |     | 1: The set value is inverted when output                             |
| 3       | _        | 0                | R   | Reserved   |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0.    |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 2   | PJ2IR    | 0                | R/W | Sets whether or not the set value is inverted when it is output on a corresponding pin. |
|     |          |                  |     | 0: The set value is not inverted when output  |
|     |          |                  |     | 1: The set value is inverted when output  |
| 1   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.                       |
| 0   | PJ0IR    | 0                | R/W | Sets whether or not the set value is inverted when it is output on a corresponding pin. |
|     |          |                  |     | 0: The set value is not inverted when output  |
|     |          |                  |     | 1: The set value is inverted when output  |

### 23.11.4 Port J Driving Ability Setting Register (PJDSR)

PJDSR is a 16-bit readable/writable register that selects the driving ability for port J. Bits PJ8DSR, PJ7DSR, PJ5DSR, PJ4DSR, PJ2DSR, and PJ0DSR correspond to pins PJ8/TxD\_B, PJ7/SCK\_B/ADEND\_A/TIJ1, PJ5/TxD\_A, PJ4/SCK\_A/ADEND\_B/TIJ0, PJ2/TxD\_A/CTx\_C/CTx\_A&CTx\_B&CTx\_C, and PJ0/TxD\_A/CTx\_A/CTx\_A&CTx\_B, respectively. The PJDSR setting is always valid regardless of the selected pin functions.

Setting the specific bits of PJDSR to 1 increases the driving ability of their corresponding pins.

PJDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. PJDSR is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8          | 7          | 6 | 5          | 4          | 3 | 2          | 1 | 0          |
|----------------|----|----|----|----|----|----|---|------------|------------|---|------------|------------|---|------------|---|------------|
|                | -  | -  | -  | -  | -  | -  | - | PJ8<br>DSR | PJ7<br>DSR | ı | PJ5<br>DSR | PJ4<br>DSR | - | PJ2<br>DSR | - | PJ0<br>DSR |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0          | 0          | 0 | 0          | 0          | 0 | 0          | 0 | 0          |
| R/W⋅           | R  | R  | R  | R  | R  | R  | R | R/M        | R/M        | R | R/M        | R/M        | R | R/W        | R | R/M        |

SH72546R Section 23 I/O Ports

| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions   |
|---------|----------|------------------|-----|--|
| 15 to 9 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 8       | PJ8DSR   | 0                | R/W | When these bits are set to 1, the driving ability of the   |
| 7       | PJ7DSR   | 0                | R/W | corresponding pin is higher than normal.   |
|         |          |                  |     | 0: Normal driving ability (slow slew rate)   |
|         |          |                  |     | 1: High driving ability (fast slew rate)   |
|         |          |                  |     | Note: For the characteristics of the driving ability, refer to section 32.3.14, Output Slew Rate.  |
| 6       | _        | 0                | R   | Reserved   |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 5       | PJ5DSR   | 0                | R/W | When these bits are set to 1, the driving ability of the   |
| 4       | PJ4DSR   | 0                | R/W | corresponding pin is higher than normal.   |
|         |          |                  |     | 0: Normal driving ability (slow slew rate)   |
|         |          |                  |     | 1: High driving ability (fast slew rate)   |
|         |          |                  |     | Note: For the characteristics of the driving ability, refer to section 32.3.14, Output Slew Rate.  |
| 3       | _        | 0                | R   | Reserved   |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 2       | PJ2DSR   | 0                | R/W | When this bit is set to 1, the driving ability of the corresponding pin is higher than normal.   |
|         |          |                  |     | 0: Normal driving ability (slow slew rate)   |
|         |          |                  |     | 1: High driving ability (fast slew rate)   |
|         |          |                  |     | Note: For the characteristics of the driving ability, refer to section 32.3.14, Output Slew Rate.  |
| 1       | _        | 0                | R   | Reserved   |
|         |          |                  |     | This bit is always read as 0. The write value should always be 0.  |
| 0       | PJ0DSR   | 0                | R/W | When this bit is set to 1, the driving ability of the corresponding pin is higher than normal.   |
|         |          |                  |     | 0: Normal driving ability (slow slew rate)   |
|         |          |                  |     | 1: High driving ability (fast slew rate)   |
|         |          |                  |     | Note: For the characteristics of the driving ability, refer to section 32.3.14, Output Slew Rate.  |
| 0       | PJ0DSR   | 0                | R/W | corresponding pin is higher than normal.  0: Normal driving ability (slow slew rate)  1: High driving ability (fast slew rate)  Note: For the characteristics of the driving ability |

### 23.11.5 Port J Pin State Setting Register (PJPSR)

PJPSR is a 16-bit readable/writable register that enables or disables the pull-down resistor of port J. Bits PJ9PSR, PJ6PSR, PJ3PSR, and PJ1PSR correspond to pins PJ9/RxD\_B, PJ6/RxD\_A, PJ3/RxD\_A/CRx\_C/CRx\_A&CRx\_B&CRx\_C, and PJ1/RxD\_A/CRx\_A/CRx\_A&CRx\_B, respectively. The PJPSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PJPSR to 1 puts their corresponding pins into pull-down state.

PJPSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9          | 8 | 7 | 6          | 5 | 4 | 3          | 2 | 1          | 0 |
|----------------|----|----|----|----|----|----|------------|---|---|------------|---|---|------------|---|------------|---|
|                | -  | 1  | -  | -  | -  | 1  | PJ9<br>PSR | - | - | PJ6<br>PSR | - | - | PJ3<br>PSR | - | PJ1<br>PSR | - |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0 | 0 | 0          | 0 | 0 | 0          | 0 | 0          | 0 |
| R/W·           | R  | R  | R  | R  | R  | R  | R/W        | R | R | R/W        | R | R | R/W        | R | R/W        | R |

| Bit      | Bit Name | Initial<br>Value | R/W | Descriptions   |
|----------|----------|------------------|-----|--|
| 15 to 10 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.             |
| 9        | PJ9PSR   | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|          |          |                  |     | 0: Pull-down resistor disabled   |
|          |          |                  |     | 1: Pull-down resistor enabled  |
| 8, 7     | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.             |
| 6        | PJ6PSR   | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|          |          |                  |     | 0: Pull-down resistor disabled   |
|          |          |                  |     | 1: Pull-down resistor enabled  |
| 5, 4     | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.             |

SH72546R Section 23 I/O Ports

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions   |
|-----|----------|------------------|-----|--|
| 3   | PJ3PSR   | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|     |          |                  |     | 0: Pull-down resistor disabled   |
|     |          |                  |     | 1: Pull-down resistor enabled  |
| 2   | _        | 0                | R   | Reserved   |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.                |
| 1   | PJ1PSR   | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|     |          |                  |     | 0: Pull-down resistor disabled   |
|     |          |                  |     | 1: Pull-down resistor enabled  |
| 0   | _        | 0                | R   | Reserved   |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.                |

### 23.12 Port K

Port K is an input/output port with the 12 pins shown in figure 23.10.

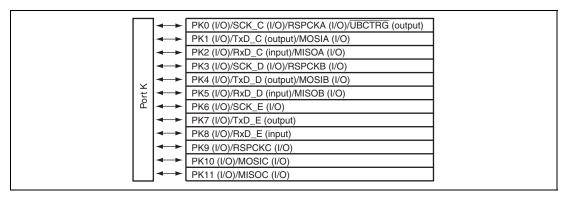


Figure 23.10 Port K

### 23.12.1 Port K Data Register (PKDR)

PKDR is a 16-bit readable/writable register that stores port K data. Bits PK11DR to PK0DR correspond to pins PK11/MISOC to PK0/SCK\_C/RSPCKA/UBCTRG, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.11 summarizes the PKDR read and write operations.

PKDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11         | 10         | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|----|----|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | -  | -  | -  | -  | PK11<br>DR | PK10<br>DR | PK9DR | PK8DR | PK7DR | PK6DR | PK5DR | PK4DR | PK3DR | PK2DR | PK1DR | PK0DR |
| Initial value: | 0  | 0  | 0  | 0  | 0          | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R  | R  | R  | R  | R/W        | R/W        | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |

SH72546R Section 23 I/O Ports

| Bit      | Bit Name | Initial<br>Value | R/W | Descriptions   |
|----------|----------|------------------|-----|--|
| 15 to 12 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 11       | PK11DR   | 0                | R/W | Refer to table 23.11.  |
| 10       | PK10DR   | 0                | R/W | _  |
| 9        | PK9DR    | 0                | R/W | _  |
| 8        | PK8DR    | 0                | R/W | _  |
| 7        | PK7DR    | 0                | R/W | _  |
| 6        | PK6DR    | 0                | R/W | _  |
| 5        | PK5DR    | 0                | R/W | _  |
| 4        | PK4DR    | 0                | R/W | _  |
| 3        | PK3DR    | 0                | R/W | _  |
| 2        | PK2DR    | 0                | R/W | _  |
| 1        | PK1DR    | 0                | R/W | _  |
| 0        | PK0DR    | 0                | R/W |  |

# Table 23.11 Read and Write Operations of Port K Data Register (PKDR)

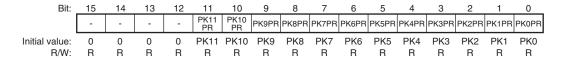
• Bits 11 to 0 in the PKDR register

| PKIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | Pin state  | Data can be written to PKDR but data does not affect the pin state. |
|       | Other than general input  | Pin state  | Data can be written to PKDR but data does not affect the pin state. |
| 1     | General output            | PKDR value | Data written to PKDR is output on the pin.                          |
|       | Other than general output | PKDR value | Data can be written to PKDR but data does not affect the pin state. |

### 23.12.2 Port K Port Register (PKPR)

PKPR is a 16-bit read-only register that always stores port K pin states. This register cannot be directly written by the CPU. The PK11PR to PK0PR bits correspond to the PK11/MISOC to PK0/SCK\_C/RSPCKA/UBCTRG pins, respectively.

When PKPR is read, the pin state can be read.



| Bit      | Bit Name | Initial<br>Value | R/W | Descriptions   |
|----------|----------|------------------|-----|--|
| 15 to 12 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 11       | PK11PR   | Pin state        | R   | When these bits are read, the pin states can be read.                |
| 10       | PK10PR   | Pin state        | R   | These bits cannot be modified.                                       |
| 9        | PK9PR    | Pin state        | R   | _  |
| 8        | PK8PR    | Pin state        | R   | _  |
| 7        | PK7PR    | Pin state        | R   | _  |
| 6        | PK6PR    | Pin state        | R   | _  |
| 5        | PK5PR    | Pin state        | R   | _  |
| 4        | PK4PR    | Pin state        | R   | _  |
| 3        | PK3PR    | Pin state        | R   | _  |
| 2        | PK2PR    | Pin state        | R   | _  |
| 1        | PK1PR    | Pin state        | R   | _  |
| 0        | PK0PR    | Pin state        | R   | _  |

SH72546R Section 23 I/O Ports

# 23.12.3 Port K Inverting Register (PKIR)

PKIR is a 16-bit readable/writable register that enables or disables the inverting function for port K. Bits PK11IR to PK9IR and PK7IR to PK0IR correspond to pins PK11/MISOC to PK9/RSPCKC and PK7/TxD\_E to PK0/SCK\_C/RSPCKA/UBCTRG, respectively. The PKIR setting is valid regardless of the port K pin functions.

Setting the specific bits of PKIR to 1 inverts the output value on their corresponding pins.

PKIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. PKIR is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13     | 12     | 11       | 10       | 9        | 8      | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|----------------|----|----|--------|--------|----------|----------|----------|--------|----------|----------|----------|----------|----------|----------|----------|----------|
|                | -  | -  | -      | -      | PK11IR   | PK10IR   | PK9IR    | -      | PK7IR    | PK6IR    | PK5IR    | PK4IR    | PK3IR    | PK2IR    | PK1IR    | PK0IR    |
| Initial value: | 0  | 0  | 0<br>B | 0<br>B | 0<br>B/W | 0<br>R/W | 0<br>B/W | 0<br>B | 0<br>B/W | 0<br>B/W | 0<br>B/W | 0<br>R/W | 0<br>R/W | 0<br>B/W | 0<br>R/W | 0<br>B/W |

| Bit      | Bit Name | Initial<br>Value | R/W | Descriptions   |
|----------|----------|------------------|-----|--|
| 15 to 12 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 11       | PK11IR   | 0                | R/W | These bits set whether or not the set value is inverted              |
| 10       | PK10IR   | 0                | R/W | when it is output on a corresponding pin.                            |
| 9        | PK9IR    | 0                | R/W | 0: The set value is not inverted when output                         |
|          |          |                  |     | 1: The set value is inverted when output                             |
| 8        | _        | 0                | R   | Reserved   |
|          |          |                  |     | This bit is always read as 0. The write value should always be 0.    |
| 7        | PK7IR    | 0                | R/W | These bits set whether or not the set value is inverted              |
| 6        | PK6IR    | 0                | R/W | when it is output on a corresponding pin.                            |
| 5        | PK5IR    | 0                | R/W | −0: The set value is not inverted when output                        |
| 4        | PK4IR    | 0                | R/W | -1: The set value is inverted when output                            |
| 3        | PK3IR    | 0                | R/W | _  |
| 2        | PK2IR    | 0                | R/W | _  |
| 1        | PK1IR    | 0                | R/W | _  |
| 0        | PK0IR    | 0                | R/W |  |

### 23.12.4 Port K Driving Ability Setting Register (PKDSR)

. ... .

PKDSR is a 16-bit readable/writable register that selects the driving ability for port K. Bits PK11DSR to PK9DSR and PK7DSR to PK0DSR correspond to pins PK11/MISOC to PK9/RSPCKC and PK7/TxD\_E to PK0/SCK\_C/RSPCKA/UBCTRG, respectively. The PKDSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PKDSR to 1 increases the driving ability of their corresponding pins.

PKDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11          | 10          | 9          | 8 | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|----|----|----|----|-------------|-------------|------------|---|------------|------------|------------|------------|------------|------------|------------|------------|
|                | -  | -  | -  | -  | PK11<br>DSR | PK10<br>DSR | PK9<br>DSR | - | PK7<br>DSR | PK6<br>DSR | PK5<br>DSR | PK4<br>DSR | PK3<br>DSR | PK2<br>DSR | PK1<br>DSR | PK0<br>DSR |
| Initial value: | 0  | 0  | 0  | 0  | 0           | 0           | 0          | 0 | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W:           | R  | R  | R  | R  | R/W         | R/W         | R/W        | R | R/W        |

|          |          | Initial |     |   |
|----------|----------|---------|-----|---|
| Bit      | Bit Name | Value   | R/W | Descriptions  |
| 15 to 12 | _        | All 0   | R   | Reserved  |
|          |          |         |     | These bits are always read as 0. The write value should always be 0.  |
| 11       | PK11DSR  | 0       | R/W | When these bits are set to 1, the driving ability of the  |
| 10       | PK10DSR  | 0       | R/W | corresponding pin is higher than normal.  |
| 9        | PK9DSR   | 0       | R/W | <ul><li>0: Normal driving ability (slow slew rate)</li></ul>  |
|          |          |         |     | 1: High driving ability (fast slew rate)  |
|          |          |         |     | Note: For the characteristics of the driving ability, refer to section 32.3.14, Output Slew Rate.                         |
| 8        | _        | 0       | R   | Reserved  |
|          |          |         |     | This bit is always read as 0. The write value should always be 0.   |
| 7        | PK7DSR   | 0       | R/W | When these bits are set to 1, the driving ability of the  |
| 6        | PK6DSR   | 0       | R/W | corresponding pin is higher than normal.  |
| 5        | PK5DSR   | 0       | R/W | Normal driving ability (slow slew rate)   |
| 4        | PK4DSR   | 0       | R/W | -1: High driving ability (fast slew rate)   |
| 3        | PK3DSR   | 0       | R/W | <ul> <li>Note: For the characteristics of the driving ability, refer<br/>to section 32.3.14, Output Slew Rate.</li> </ul> |
| 2        | PK2DSR   | 0       | R/W | _ '   |
| 1        | PK1DSR   | 0       | R/W | _   |
| 0        | PK0DSR   | 0       | R/W |   |
|          |          |         |     |   |

SH72546R Section 23 I/O Ports

# 23.12.5 Port K Pin State Setting Register (PKPSR)

PKPSR is a 16-bit readable/writable register that enables or disables the pull-down resistor of port K. Bits PK11PSR, PK8PSR, PK5PSR, and PK2PSR correspond to pins PK11/MISOC, PK8/RxD\_E, PK5/RxD\_D/MISOB, and PK2/RxD\_C/MISOA, respectively. The PKPSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PKPSR to 1 puts their corresponding pins into pull-down state.

PKPSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11          | 10 | 9 | 8          | 7 | 6 | 5          | 4 | 3 | 2          | 1 | 0 |
|----------------|----|----|----|----|-------------|----|---|------------|---|---|------------|---|---|------------|---|---|
|                | -  | -  | -  | -  | PK11<br>PSR | -  | - | PK8<br>PSR | - | - | PK5<br>PSR | - | - | PK2<br>PSR | - | - |
| Initial value: | 0  | 0  | 0  | 0  | 0           | 0  | 0 | 0          | 0 | 0 | 0          | 0 | 0 | 0          | 0 | 0 |
| R/W:           | R  | R  | R  | R  | R/W         | R  | R | R/W        | R | R | R/W        | R | R | R/W        | R | R |

| Bit      | Bit Name | Initial<br>Value | R/W | Descriptions   |
|----------|----------|------------------|-----|--|
| 15 to 12 | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.             |
| 11       | PK11PSR  | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|          |          |                  |     | 0: Pull-down resistor disabled   |
|          |          |                  |     | 1: Pull-down resistor enabled  |
| 10, 9    | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.             |
| 8        | PK8PSR   | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|          |          |                  |     | 0: Pull-down resistor disabled   |
|          |          |                  |     | 1: Pull-down resistor enabled  |
| 7, 6     | _        | All 0            | R   | Reserved   |
|          |          |                  |     | These bits are always read as 0. The write value should always be 0.             |

| Bit  | Bit Name | Initial<br>Value | R/W | Descriptions   |
|------|----------|------------------|-----|--|
| 5    | PK5PSR   | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|      |          |                  |     | 0: Pull-down resistor disabled   |
|      |          |                  |     | 1: Pull-down resistor enabled  |
| 4, 3 | _        | All 0            | R   | Reserved   |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.             |
| 2    | PK2PSR   | 0                | R/W | Selects whether or not the pull-down resistor of a corresponding pin is enabled. |
|      |          |                  |     | 0: Pull-down resistor disabled   |
|      |          |                  |     | 1: Pull-down resistor enabled  |
| 1, 0 | _        | All 0            | R   | Reserved   |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.             |

SH72546R Section 23 I/O Ports

#### 23.13 Port L

Port L is an input/output port with the 9 pins shown in figure 23.11.

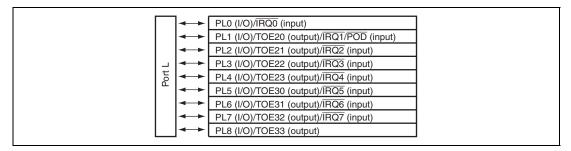


Figure 23.11 Port L

#### 23.13.1 Port L Data Register (PLDR)

PLDR is a 16-bit readable/writable register that stores port L data. Bits PL8DR to PL0DR correspond to pins PL8/TOE33 to PL0/IRQ0, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 23.12 summarizes the PLDR read and write operations.

PLDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|----------------|----|----|----|----|----|----|---|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | 1  | -  | -  | -  | -  | 1  | - | PL8DR | PL7DR | PL6DR | PL5DR | PL4DR | PL3DR | PL2DR | PL1DR | PL0DR |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R/W   |

| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions   |
|---------|----------|------------------|-----|--|
| 15 to 9 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 8       | PL8DR    | 0                | R/W | Refer to table 23.12.  |
| 7       | PL7DR    | 0                | R/W | _  |
| 6       | PL6DR    | 0                | R/W | _  |
| 5       | PL5DR    | 0                | R/W | _  |
| 4       | PL4DR    | 0                | R/W | _  |
| 3       | PL3DR    | 0                | R/W | _  |
| 2       | PL2DR    | 0                | R/W | _  |
| 1       | PL1DR    | 0                | R/W | _  |
| 0       | PL0DR    | 0                | R/W | _  |

# Table 23.12 Read and Write Operations of Port L Data Register (PLDR)

• Bits 8 to 0 in the PLDR register

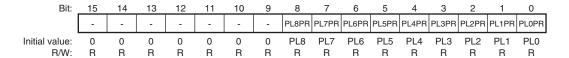
| PLIOR | Pin Function              | Read       | Write   |
|-------|---------------------------|------------|---|
| 0     | General input             | Pin state  | Data can be written to PLDR but data does not affect the pin state. |
|       | Other than general input  | Pin state  | Data can be written to PLDR but data does not affect the pin state. |
| 1     | General output            | PLDR value | Data written to PLDR is output on the pin.                          |
|       | Other than general output | PLDR value | Data can be written to PLDR but data does not affect the pin state. |

SH72546R Section 23 I/O Ports

## 23.13.2 Port L Port Register (PLPR)

PLPR is a 16-bit read-only register that always stores port L pin states. This register cannot be directly written by the CPU. Bits PL8PR to PL0PR correspond to pins PL8/TOE33 to  $PL0/\overline{IRQ0}$ , respectively.

When PLPR is read, the pin state can be read.



| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions   |
|---------|----------|------------------|-----|--|
| 15 to 9 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 8       | PL8PR    | Pin state        | R   | When these bits are read, the pin states can be read.                |
| 7       | PL7PR    | Pin state        | R   | These bits cannot be modified.                                       |
| 6       | PL6PR    | Pin state        | R   | _  |
| 5       | PL5PR    | Pin state        | R   | <del>-</del>   |
| 4       | PL4PR    | Pin state        | R   | <del>-</del>   |
| 3       | PL3PR    | Pin state        | R   | _  |
| 2       | PL2PR    | Pin state        | R   | _  |
| 1       | PL1PR    | Pin state        | R   | _  |
| 0       | PL0PR    | Pin state        | R   | _  |

# 23.13.3 Port L Inverting Register (PLIR)

PLIR is a 16-bit readable/writable register that enables or disables the inverting function for port L. Bits PL8IR to PL1IR correspond to pins PL8/TOE33 to PL1/TOE20/IRQ1/POD, respectively. The PLIR setting is valid regardless of the port L pin functions.

Setting the specific bits of PLIR to 1 inverts the output value on their corresponding pins.

PLIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0 |
|----------------|----|----|----|----|----|----|---|-------|-------|-------|-------|-------|-------|-------|-------|---|
|                | -  | -  | -  | -  | -  | -  | - | PL8IR | PL7IR | PL6IR | PL5IR | PL4IR | PL3IR | PL2IR | PL1IR | - |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0 |
| R/W:           | R  | R  | R  | R  | R  | R  | R | R/W   | R |

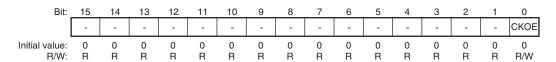
|         |          | Initial |     |  |
|---------|----------|---------|-----|--|
| Bit     | Bit Name | Value   | R/W | Descriptions   |
| 15 to 9 | _        | All 0   | R   | Reserved   |
|         |          |         |     | These bits are always read as 0. The write value should always be 0. |
| 8       | PL8IR    | 0       | R/W | These bits set whether or not the set value is inverted              |
| 7       | PL7IR    | 0       | R/W | when it is output on a corresponding pin.                            |
| 6       | PL6IR    | 0       | R/W | ─ 0: The set value is not inverted when output                       |
| 5       | PL5IR    | 0       | R/W | -1: The set value is inverted when output                            |
| 4       | PL4IR    | 0       | R/W | _  |
| 3       | PL3IR    | 0       | R/W | _  |
| 2       | PL2IR    | 0       | R/W | <del>-</del>   |
| 1       | PL1IR    | 0       | R/W | <del>-</del>   |
| 0       | _        | 0       | R   | Reserved   |
|         |          |         |     | This bit is always read as 0. The write value should always be 0.    |

SH72546R Section 23 I/O Ports

# **CK Control Register (CKCR)**

CKCR is a 16-bit readable/writable register that enables or disables the CK output.

CKCR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.



| Bit     | Bit Name | Initial<br>Value | R/W | Descriptions   |
|---------|----------|------------------|-----|--|
| 15 to 1 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0. |
| 0       | CKOE     | 0                | R/W | Enables or disables an output on the CK pin.                         |
|         |          |                  |     | 0: Enables output on the pin (CK output)                             |
|         |          |                  |     | 1: Disables output on the pin (Hi-Z)                                 |

# 23.15 Port Output Disable (POD)

The output buffers of I/O port related pins (shown in table 23.13) can be controlled by the input level on the  $\overline{POD}$  (port output disable) pin. This function is enabled regardless of the selected function when the corresponding pin is specified as an output. The output buffer control by the  $\overline{POD}$  pin is performed asynchronously with the bus cycles.

### Table 23.13 I/O Port Relating Pins

### **Relating Pins**

PB11/CS1/TOE21, PB8/WAIT/TOE20,
PD12/TCLKA/TIOC41/TIJ0 to PD0/ TIOC00/TIOC31,
PE13/TOE13 to PE6/TOE00/CTx\_B, PE2/TIA02/TIOC43/TIOC30, PE1/TIA01/TIOC42/TIOC40,
PF15/TOD33B/CRx\_B/RxD\_A to PF0/TOD00B/TIF6,
PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0,
PL8/TOE33 to PL2/TOE21/IRQ2

### Table 23.14 POD Pin State

| POD | Descriptions                                      |
|-----|---|
| 0   | The pin output is disabled (high impedance)       |
| 1   | The pin output is enabled (each setting function) |

SH72546R Section 23 I/O Ports

#### 23.16 **Usage Note**

#### 23,16,1 **Note on State Immediately after Reset**

Immediately after reset, input/output pins are in the input state. In this case, if the pins are at intermediate levels because, for example, they are opened, a passthrough current may flow. To reduce the passthrough current, set general input/output pins in the output state at low level, immediately after the LSI has been turned on. This countermeasure will set the input/output pins stable, and therefore the passthrough current will be reduced.

#### 23.16.2 Note on Operation of Input/Output Pins on a Reset by an Internal Source

When input/output pins enter the reset state from the operating state due to a reset by an internal source, the states of the pins are undefined for up to one cycle of the peripheral clock ( $P\phi$ ). Over this period, the pins may be in the high-level output state, the low-level output state, or the highimpedance state.

Although the input/output pins are only temporarily in the undefined state described above, the microcontroller as a whole enters the specified reset state after that.

Resets by internal sources include internal resets by WDT overflow and H-UDI resets using emulators.

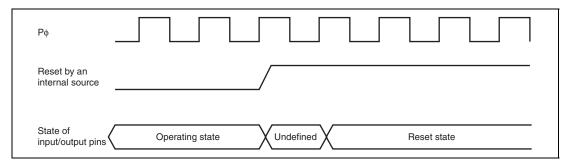


Figure 23.12 Operation of Input/Output Pins at Reset by Internal Source

The phenomenon described above does not occur on any of dedicated input or output pins. Note:

Resets by inputting a low level on the RES pin do not correspond the resets by internal sources.

# Section 24 Multi-Input Signature Generator (MISG)

#### 24.1 Overview

This LSI has a multi-input signature generator (MISG) for self test function.

The MISG monitors any write accesses by the CPU to the calculation data register (MISRCDR) and generates a 32-bit signature using the contents of MISRCDR. The signature is stored in the multi-input signature register (MISR). Each time MISRCDR is accessed, the MISG updates the contents of MISR with a new signature which is generated by the data written to MISRCDR and the data stored in MISR (normally the data in MISR is the signature generated by the previous write access; in some cases, it is an initial value or a value that is directly written to MISR).

Signatures are generated by the following polynomial.

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

The enable bit in the MISR control register (MISRCR) can be used to enable/disable the generation of signatures. In the initial status after a reset, signature generation is disabled so that writing to MISRCDR will not update the data stored in MISR.

MISR is a register that is address-mapped to the internal I/O register space and can be directly read from or written to. A generated signature in MISR can be compared with an expected value or an arbitrary seed value (the value that is the source of a signature) can also be stored.

Figure 24.1 is a block diagram of the MISG.

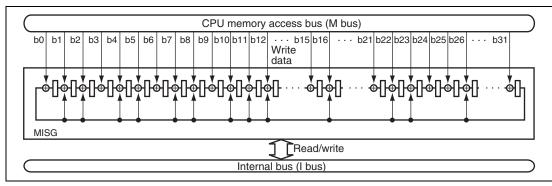


Figure 24.1 Block Diagram of MISG

# 24.2 Register Descriptions

The MISG has the following registers.

**Table 24.1 Register Configuration** 

| Register Name                  | Symbol  | R/W | Initial Value | Address    | Access Size |
|--------------------------------|---------|-----|---------------|------------|-------------|
| Calculation data register      | MISRCDR | W   | _             | H'FFF7FFFC | 8, 16, 32   |
| MISR control register          | MISRCR  | R/W | H'00          | H'FFFC1C00 | 8           |
| Multi-input signature register | MISR    | R/W | H'00000000    | H'FFFC1C04 | 32          |

### 24.2.1 Calculation Data Register (MISRCDR)

MISRCDR is a 32-bit write-only register. MISRCDR can be written only by the CPU. MISRCDR can always be accessed in one write cycle (CPUCLK). The data written to MISRCDR is used as input data for the multi-input signature register (MISR).

Data can be written to MISRCDR in words, bytes, or longwords. However, the word or longword data is extended to 32-bit data by filling remaining bits with 0 so that 32-bit data is always input into MISR.

Note: Do not read from MISRCDR. Otherwise, correct operation cannot be guaranteed.

| Bit:           | 31 | 30            | 29 | 28 | 27 | 26 | 25 | 24    | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|----|---------------|----|----|----|----|----|-------|---------|----|----|----|----|----|----|----|
|                |    | MISRCD[31:16] |    |    |    |    |    |       |         |    |    |    |    |    |    |    |
| Initial value: | -  | -             | -  | -  | -  | -  | -  | -     | -       | -  | -  | -  | -  | -  | -  | -  |
| R/W:           | W  | W             | W  | W  | W  | W  | W  | W     | W       | W  | W  | W  | W  | W  | W  | W  |
| 5              |    |               |    |    |    |    | _  | _     | _       | _  | _  |    | _  | _  |    |    |
| Bit:           | 15 | 14            | 13 | 12 | 11 | 10 | 9  | 8     | /       | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|                |    |               |    |    |    |    |    | MISRC | D[15:0] |    |    |    |    |    |    |    |
| Initial value: | -  | -             | -  | -  | -  | -  | -  | -     | -       | -  | -  | -  | -  | -  | -  | -  |
| R/W:           | W  | W             | W  | W  | W  | W  | W  | W     | W       | W  | W  | W  | W  | W  | W  | W  |

| Bit     | Bit Name | Initial<br>Value | R/W | Description   |
|---------|----------|------------------|-----|---|
| 31 to 0 | MISRCD   | _                | W   | Calculation Data  |
|         | [31:0]   |                  |     | Input data to MISR. Each time calculation data is written, a new signature is generated and it is stored in MISR. |

### 24.2.2 Multi-Input Signature Register (MISR)

MISR is a 32-bit readable/writable register. Setting the MISR enable bit (MISREN) in the MISR control register (MISRCR) to 1 causes MISR to generate a new signature whenever data is written to MISRCDR, and the generated value is stored in MISR.

Signatures are generated by the following polynomial.

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

MISR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Note: When writing to MISRCDR after writing to this register, read this register before writing to MISRCDR. If MISR is not read, a correct signature may not be generated through a write operation to MISRCDR.

If the interval between writing to the MISRCDR register and reading from the MISR register is too short when four-times and two-times multiplication have been specified for the internal clock ( $\phi$ ) and peripheral clock ( $P\phi$ ), respectively, a signature generated by writing to MISRCDR may not be read correctly. If it is necessary to read from the MISR register immediately after writing to the MISRCDR register, execute three or more instructions before reading from the MISR register or read twice from the MISR register and use the second read data.

| Bit:                   | 31         | 30          | 29       | 28       | 27       | 26       | 25       | 24       | 23       | 22       | 21       | 20       | 19       | 18       | 17       | 16       |
|------------------------|------------|-------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|                        |            | MISR[31:16] |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | 0<br>R/W   | 0<br>R/W    | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W |
| Bit:                   | 15         | 14          | 13       | 12       | 11       | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
|                        | MISR[15:0] |             |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Initial value:<br>R/W: | 0<br>R/W   | 0<br>R/W    | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W | 0<br>R/W |

| Bit     | Bit Name    | Initial<br>Value | R/W | Description                                       |
|---------|-------------|------------------|-----|---|
| 31 to 0 | MISR [31:0] | H'00000000       | R/W | Signature Data                                    |
|         |             |                  |     | These bits return the latest signature when read. |

### 24.2.3 MISR Control Register (MISRCR)

MISRCR is an 8-bit readable/writable register that controls the generation of a signature by MISR. MISR generates a signature when the MISR enable bit (MISREN) is set to 1 and stores the resulting signature. If the MISREN bit is 0, the contents of MISR are not updated even when data is written to MISRCDR.

MISRCR is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Note: When writing to MISRCDR after writing to this register, perform either of the following two operations before writing to MISRCDR. A correct signature may not be generated through a write operation to MISRCDR without performing either of them.

- 1 Read twice from the MISRCR register.
- 2 Execute three or more instructions after reading from the MISRCR register.

| Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
|----------------|---|---|---|---|---|---|---|--------|
|                | - | - | - | - | - | - | - | MISREN |
| 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<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7 to 1 | _        | All 0            | R   | Reserved  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 0      | MISREN   | 0                | R/W | MISR Enable   |
|        |          |                  |     | <ol> <li>Disables generating a signature or updating the<br/>contents of MISR even when data is written to<br/>MISRCDR</li> </ol> |
|        |          |                  |     | <ol> <li>Enables generating a signature or updating the<br/>contents of MISR even when data is written to<br/>MISRCDR</li> </ol>  |

SH72546R Section 25 ROM

# Section 25 ROM

This LSI incorporates 3.75 Mbytes of flash memory (ROM) for the storage of instruction code. The flash memory has the following features.

### 25.1 Features

Two types of flash-memory MATs

The ROM has two types of memory areas (hereafter referred to as memory MATs) in the same address space. These two MATs can be switched by the start-up mode or bank switching through the control register. For addresses H'00008000 to H'003BFFFF, undefined data is read and programming and erasing are ignored when the user boot MAT is selected.

User MAT: 3.75 Mbytes User boot MAT: 32 Kbytes

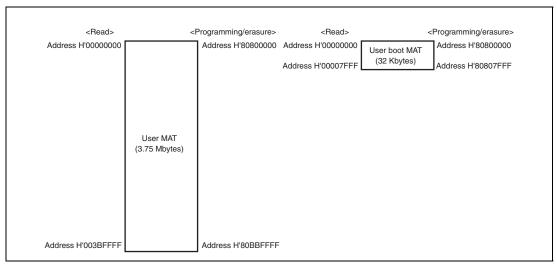


Figure 25.1 Memory MAT Configuration in ROM

- High-speed reading through ROM cache
   Both the user MAT and user boot MAT can be read at high speed through the ROM cache.
   They can be read only in on-chip ROM enabled mode.
- Programming and erasing methods
   The ROM can be programmed and erased by commands issued through the peripheral bus (P bus) to the ROM/EEPROM-dedicated sequencer (FCU).

Section 25 ROM SH72546R

While the flash control unit (FCU) is programming or erasing the ROM, the CPU can execute a program located outside the ROM. While the FCU is programming or erasing the EEPROM, the CPU can execute a program in the ROM. When the FCU suspends programming or erasure, the CPU can execute a program in the ROM, and then the FCU can resume programming or erasure. While the FCU suspends erasure, areas other than the erasure-suspended area can be programmed.

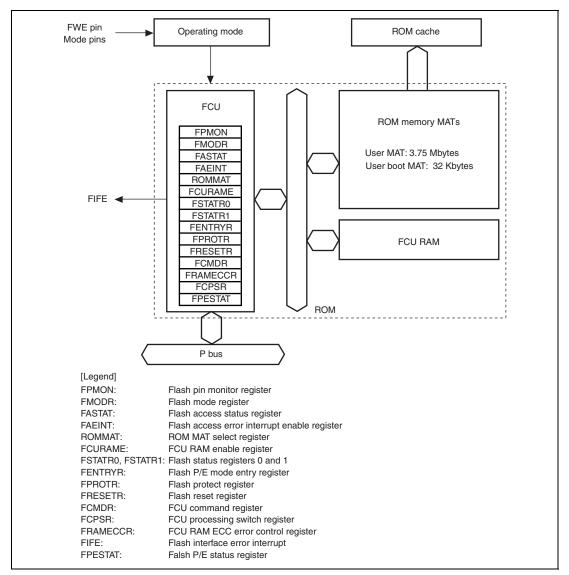


Figure 25.2 Block Diagram of ROM

SH72546R Section 25 ROM

### Programming/erasing unit

The user MAT and user boot MAT are programmed in 256-byte units. The entire area of the user boot MAT is always erased at one time.

Figure 25.3 shows the block configuration of the user MAT. The user MAT is divided into eight 8-Kbyte blocks, nine 64-Kbyte blocks, and twenty-five 128-Kbyte blocks.

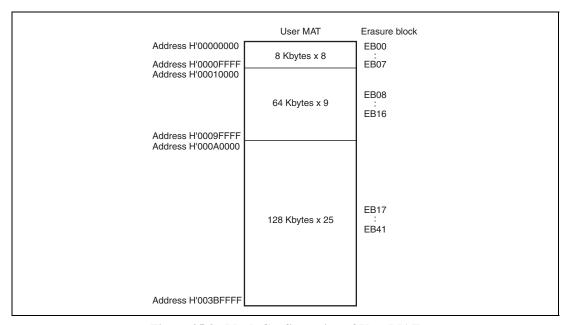


Figure 25.3 Block Configuration of User MAT

Section 25 ROM SH72546R

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

The user MAT and user boot MAT can be programmed using the SCI. The bit rate for SCI communications between the host and this LSI can be automatically adjusted.

— User program mode

The user MAT can be programmed with a desired interface. A transition from MCU mode 2 (MCU extended mode) or mode 3 (MCU single-chip mode) to this mode is enabled simply by changing the level on the FWE pin.

User boot mode

The user MAT can be programmed with a desired interface. To make a transition to this mode, a reset is needed.

Protection modes

This LSI supports two modes to protect memory against programming or erasure: hardware protection by the levels on the FWE and mode pins and software protection by the FENTRY5 to FENTRY0 bits or lock bit settings. The FENTRY5 to FENTRY0 bits enable or disable ROM programming or erasure by the FCU. A lock bit is included in each erasure block of the user MAT to protect memory against programming or erasure.

The LSI also provides a function to suspend programming or erasure when abnormal operation is detected during programming or erasure.

Programming and erasing time and count
 Refer to section 32, Electrical Characteristics.

SH72546R Section 25 ROM

# 25.2 Input/Output Pins

Table 25.1 shows the input/output pins used for the ROM. The combination of MD4 to MD0 pin levels and the FWE pin level determines the ROM programming mode (see section 25.4, Overview of ROM-Related Modes). In boot mode, the ROM can be programmed or erased by the host connected via the PJ6/RxD\_A and PJ5/TxD\_A pins (see section 25.5, Boot Mode).

**Table 25.1 Pin Configuration** 

| Pin Name                       | Symbol     | I/O    | Function  |
|--------------------------------|------------|--------|---|
| Power-on reset                 | RES        | Input  | This LSI enters the power-on reset state when this signal goes low. |
| Mode                           | MD4 to MD0 | Input  | These pins specify the operating mode.                              |
| Flash programming enable       | FWE        | Input  | This pin enables or disables ROM programming.                       |
| Receive data in SCI channel A  | PJ6/RxD_A  | Input  | Receives data through SCI channel A (communications with host)      |
| Transmit data in SCI channel A | PJ5/TxD_A  | Output | Transmits data through SCI channel A (communications with host)     |

Section 25 ROM SH72546R

# 25.3 Register Descriptions

Table 25.2 shows the ROM-related registers. Some of these registers have EEPROM-related bits, but this section only describes the ROM-related bits. For the EEPROM-related bits, refer to section 26.3, Register Descriptions. The ROM-related registers are initialized by a power-on reset or a transition to the hardware standby mode.

Initial

**Table 25.2 Register Configuration** 

| Register Name                                | Symbol   | R/W* <sup>1</sup>   | Initial<br>Value   | Address      | Access<br>Size |
|--|----------|---------------------|--------------------|--------------|----------------|
| Flash pin monitor register                   | FPMON    | R                   | H'00               | H'FFFFA800   | 8              |
| That pir monitor register                    | TT MICH  | 11                  | H'80               | 111111111000 | Ü              |
| Floob woods we sisted                        | FMODR    | R/W                 | H'00               | H'FFFFA802   |                |
| Flash mode register                          | FINIODK  |                     | П 00               | П ГГГГА602   | <u> </u>       |
| Flash access status register                 | FASTAT   | R/(W)* <sup>2</sup> | H'00               | H'FFFFA810   | 8              |
| Flash access error interrupt enable register | FAEINT   | R/W                 | H'9F               | H'FFFFA811   | 8              |
| ROM MAT select register                      | ROMMAT   | R/(W)*3             | H'0000             | H'FFFFA820   | 8, 16          |
|  |          |                     | H'0001             |              |                |
| FCU RAM enable register                      | FCURAME  | R/(W)*3             | H'0000             | H'FFFFA854   | 8, 16          |
| Flash status register 0                      | FSTATR0  | R                   | H'80* <sup>5</sup> | H'FFFFA900   | 8, 16          |
| Flash status register 1                      | FSTATR1  | R                   | H'00*5             | H'FFFFA901   | 8, 16          |
| Flash P/E mode entry register                | FENTRYR  | R/(W)*4             | H'0000*5           | H'FFFFA902   | 8, 16          |
| Flash protect register                       | FPROTR   | R/(W)*4             | H'0000*5           | H'FFFFA904   | 8, 16          |
| Flash reset register                         | FRESETR  | R/(W)*3             | H'0000             | H'FFFFA906   | 8, 16          |
| FCU command register                         | FCMDR    | R                   | H'FFFF*5           | H'FFFFA90A   | 8, 16          |
| FCU RAM ECC error control register           | FRAMECCR | R/W                 | H'02* <sup>5</sup> | H'FFFFA90C   | 8              |
| FCU processing switch register               | FCPSR    | R/W                 | H'0000*5           | H'FFFFA918   | 8, 16          |
| Flash P/E status register                    | FPESTAT  | R                   | H'0000*5           | H'FFFFA91C   | 8, 16          |

Notes: 1. In on-chip ROM disabled mode, the ROM-related registers are always read as 0 and writing to them is ignored.

- 2. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.
- 3. This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.

SH72546R Section 25 ROM

4. This register can be written to only when a specified value is written to the upper byte in word access; the register is initialized when a value not allowed for the register is written to the upper byte. The data written to the upper byte is not stored in the register.

5. These registers can be initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

### 25.3.1 Flash Pin Monitor Register (FPMON)

FPMON monitors the FWE pin state. FPMON is read as H'00 in on-chip ROM disabled mode. FPMON is initialized by a power-on reset or a transition to the hardware standby mode.

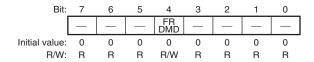
| Bit:           | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|-----|---|---|---|---|---|---|---|
|                | FWE | _ | _ | _ | _ | _ | _ | _ |
| Initial value: | 1/0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W:           | R   | R | R | R | R | R | R | R |

| D.,    | D'AN     | Initial | D 04/ | Post total  |
|--------|----------|---------|-------|---|
| Bit    | Bit Name | Value   | R/W   | Description   |
| 7      | FWE      | 1/0     | R     | Flash Write Enable  |
|        |          |         |       | Monitors the FWE pin level. The initial value depends on the FWE pin level when the LSI is started. |
|        |          |         |       | 0: Disables ROM programming and erasure   |
|        |          |         |       | 1: Enables ROM programming and erasure  |
| 6 to 0 | _        | All 0   | R     | Reserved  |
|        |          |         |       | These bits are always read as 0. The write value should always be 0.                                |

Section 25 ROM SH72546R

#### 25.3.2 Flash Mode Register (FMODR)

FMODR specifies the FCU operation mode. In on-chip ROM disabled mode, FMODR is read as H'00 and writing to it is ignored. FMODR is initialized by a power-on reset or a transition to the hardware standby mode.

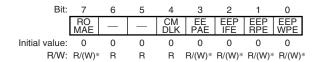


|        |          | Initial |     |   |
|--------|----------|---------|-----|---|
| Bit    | Bit Name | Value   | R/W | Description   |
| 7 to 5 | _        | All 0   | R   | Reserved  |
|        |          |         |     | These bits are always read as 0. The write value should always be 0.  |
| 4      | FRDMD    | 0       | R/W | FCU Read Mode Select  |
|        |          |         |     | Selects the read mode to read the ROM or EEPROM using FCU. This bit specifies the check method for the lock bits in the ROM (see section 25.6.1, FCU Command List, and section 25.6.3 (11), Reading Lock Bit), whereas this bit must be set to make the blank check command available for use in the EEPROM (see section 26, EEPROM). |
|        |          |         |     | 0: Selects the memory area read mode.   |
|        |          |         |     | The mode to read the lock bits in the ROM in ROM lock bit read mode.  |
|        |          |         |     | 1: Selects the register read mode.  |
|        |          |         |     | The mode to read the lock bits in the ROM using the lock bit read 2 command.  |
| 3 to 0 | _        | All 0   | R   | Reserved  |
|        |          |         |     | These bits are always read as 0. The write value should always be 0.  |

SH72546R Section 25 ROM

### 25.3.3 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and EEPROM. In on-chip ROM disabled mode, FASTAT is read as H'00 and writing to it is ignored. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 25.8.3, Error Protection). To cancel a command-locked state, set FASTAT to H'10, and then issue a status-clear command to the FCU. FASTAT is initialized by a power-on reset or a transition to the hardware standby mode.



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

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |
|-----|----------|------------------|--------|---|
| 7   | ROMAE    | 0                | R/(W)* | Access Error  |
|     |          |                  |        | Indicates whether or not a ROM access error has been generated. If this bit becomes 1, the ILGLERR bit in FSTATR0 is set to 1 and the FCU enters a command-locked state.              |
|     |          |                  |        | 0: No ROM access error has occurred.  |
|     |          |                  |        | 1: A ROM access error has occurred.   |
|     |          |                  |        | [Setting conditions]  |
|     |          |                  |        | <ul> <li>A read access is issued to ROM program/erase<br/>addresses H'80B80000 to H'80BBFFFF while the<br/>FENTRY5 bit in FENTRYR is 1 in ROM P/E<br/>normal mode.</li> </ul>         |
|     |          |                  |        | <ul> <li>A read access command is issued to ROM<br/>program/erase addresses H'80B40000 to<br/>H'80B7FFFF while the FENTRY4 bit in FENTRYR<br/>is 1 in ROM P/E normal mode.</li> </ul> |
|     |          |                  |        | <ul> <li>A read access command is issued to ROM<br/>program/erase addresses H'80B00000 to<br/>H'80B3FFFF while the FENTRY3 bit in FENTRYR<br/>is 1 in ROM P/E normal mode.</li> </ul> |

SH72546R Section 25 ROM

| Bit | Bit Name | Initial<br>Value | R/W    | Description  |
|-----|----------|------------------|--------|--|
| 7   | ROMAE    | 0                | R/(W)* | <ul> <li>A read access is issued to ROM program/erase addresses H'80A00000 to H'80AFFFFF while the FENTRY2 bit in FENTRYR is 1 in ROM P/E normal mode.</li> <li>A read access command is issued to ROM program/erase addresses H'80900000 to H'809FFFFF while the FENTRY1 bit in FENTRYR is 1 in ROM P/E normal mode.</li> <li>A read access command is issued to ROM program/erase addresses H'80800000 to H'808FFFF while the FENTRY0 bit in FENTRYR is 1 in ROM P/E normal mode.</li> <li>An access is issued to ROM program/erase addresses H'80B80000 to H'80BBFFFF while the FENTRY5 bit in FENTRYR is 0.</li> <li>An access command is issued to ROM program/erase addresses H'80B40000 to H'80B7FFFF while the FENTRY4 bit in FENTRYR is 0.</li> <li>An access command is issued to ROM program/erase addresses H'80B00000 to H'80B3FFFF while the FENTRY3 bit in FENTRYR is 0.</li> <li>An access is issued to ROM program/erase addresses H'80A00000 to H'80AFFFFF while the FENTRY2 bit in FENTRYR is 0.</li> <li>An access command is issued to ROM program/erase addresses H'80A00000 to H'80AFFFFF while the FENTRY1 bit in FENTRYR is 0.</li> <li>An access command is issued to ROM program/erase addresses H'80900000 to H'809FFFFF while the FENTRY1 bit in FENTRYR is 0.</li> <li>An access command is issued to ROM program/erase addresses H'80900000 to H'809FFFFF while the FENTRY1 bit in FENTRYR is 0.</li> <li>An access command is issued to ROM program/erase addresses H'80900000 to H'808FFFFF while the FENTRY1 bit in FENTRYR is 0.</li> </ul> |

SH72546R Section 25 ROM

| Bit  | Bit Name | Initial<br>Value | R/W    | Description   |
|------|----------|------------------|--------|---|
| 7    | ROMAE    | 0                | R/(W)* | <ul> <li>A read access command is issued to ROM read addresses H'00000000 to H'003BFFFF while the FENTRYR register value is not H'0000.</li> <li>A block erase, program, or lock bit program command is issued while the user boot MAT is selected.</li> <li>An access command is issued to an address other than ROM program/erase addresses H'80800000</li> </ul> |
|      |          |                  |        | to H'80807FFF while the user boot MAT is selected.  |
|      |          |                  |        | [Clearing condition]  |
|      |          |                  |        | A 0 is written to this bit after reading a 1 from the ROMAE bit.  |
| 6, 5 | _        | All 0            | R      | Reserved  |
|      |          |                  |        | These bits are always read as 0. The write value should always be 0.  |
| 4    | CMDLK    | 0                | R      | FCU Command Lock  |
|      |          |                  |        | Indicates whether the FCU is in command-locked state (see section 25.8.3, Error Protection).  |
|      |          |                  |        | 0: The FCU is not in a command-locked state   |
|      |          |                  |        | 1: The FCU is in a command-locked state   |
|      |          |                  |        | [Setting condition]   |
|      |          |                  |        | The FCU detects an error and enters command-  |
|      |          |                  |        | locked state.   |
|      |          |                  |        | [Clearing condition]  |
|      |          |                  |        | <ul> <li>The FCU completes the status-clear command<br/>processing while FASTAT is H'10.</li> </ul>   |
| 3    | EEPAE    | 0                | R/(W)* | EEPROM Access Error   |
|      |          |                  |        | Refer to section 26, EEPROM.  |
| 2    | EEPIFE   | 0                | R/(W)* | EEPROM Instruction Fetch Error  |
|      |          |                  |        | Refer to section 26, EEPROM.  |
| 1    | EEPRPE   | 0                | R/(W)* | EEPROM Read Protect Error   |
|      |          |                  |        | Refer to section 26, EEPROM.  |
| 0    | EEPWPE   | 0                | R/(W)* | EEPROM Program/Erase Protect Error  |
|      |          |                  |        | Refer to section 26, EEPROM.  |

Section 25 ROM SH72546R

Note: \* Writing a 0 after reading a 1 is only allowed in order to clear the flag.

# 25.3.4 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupts. In on-chip ROM disabled mode, FAEINT is read as H'00 and writing to it is ignored. FAEINT is initialized by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7           | 6 | 5 | 4           | 3           | 2            | 1            | 0            |
|----------------|-------------|---|---|-------------|-------------|--------------|--------------|--------------|
|                | ROM<br>AEIE | _ | _ | CMD<br>LKIE | EEP<br>AEIE | EEPI<br>FEIE | EEPR<br>PEIE | EEPW<br>PEIE |
| Initial value: | 1           | 0 | 0 | 1           | 1           | 1            | 1            | 1            |
| R/W·           | R/W         | R | R | R/W         | R/W         | R/W          | R/W          | R/W          |

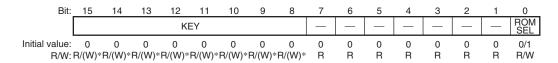
| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 7    | ROMAEIE  | 1                | R/W | ROM Access Error Interrupt Enable   |
|      |          |                  |     | Enables or disables an FIFE interrupt request when a ROM access error occurs and the ROMAE bit in FASTAT becomes 1.           |
|      |          |                  |     | 0: Does not generate an FIFE interrupt request when ROMAE = 1.  |
|      |          |                  |     | 1: Generates an FIFE interrupt request when ROMAE = 1.  |
| 6, 5 | _        | All 0            | R   | Reserved  |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 4    | CMDLKIE  | 1                | R/W | FCU Command Lock Interrupt Enable   |
|      |          |                  |     | Enables or disables an FIFE interrupt request when FCU command-locked state is entered and the CMDLK bit in FASTAT becomes 1. |
|      |          |                  |     | 0: Does not generate an FIFE interrupt request when CMDLK = 1   |
|      |          |                  |     | 1: Generates an FIFE interrupt request when CMDLK = 1   |
| 3    | EEPAEIE  | 1                | R/W | EEPROM Access Error Interrupt Enable  |
|      |          |                  |     | Refer to section 26, EEPROM.  |
| 2    | EEPIFEIE | 1                | R/W | EEPROM Instruction Fetch Error Interrupt Enable   |
|      |          |                  |     | Refer to section 26, EEPROM.  |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 1   | EEPRPEIE | 1                | R/W | EEPROM Read Protect Error Interrupt Enable             |
|     |          |                  |     | Refer to section 26, EEPROM.                           |
| 0   | EEPWPEIE | 1                | R/W | EEPROM Program/Erase Protect Error Interrupt<br>Enable |
|     |          |                  |     | Refer to section 26, EEPROM.                           |

### 25.3.5 ROM MAT Select Register (ROMMAT)

ROMMAT switches memory MATs in the ROM. In on-chip ROM disabled mode, ROMMAT is read as H'0000 and writing to it is ignored. ROMMAT is initialized by a power-on reset or a transition to the hardware standby mode.

When switching MAT by changing ROMMAT register setting, write to ROMMAT register and execute a minimum of 5 NOP instructions after performing a dummy read of ROMMAT register.



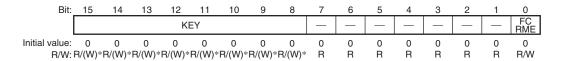
Note: \* Written data is not stored in these bits.

| Bit     | Bit Name | Initial<br>Value | R/W    | Description  |
|---------|----------|------------------|--------|--|
| 15 to 8 | KEY      | H'00             | R/(W)* | Key Code   |
|         |          |                  |        | These bits enable or disable ROMSEL bit modification. The data written to these bits are not stored.   |
| 7 to 1  | _        | All 0            | R      | Reserved   |
|         |          |                  |        | These bits are always read as 0. The write value should always be 0.   |
| 0       | ROMSEL   | 0/1              | R/W    | ROM MAT Select   |
|         |          |                  |        | Selects a memory MAT in the ROM. The initial value is 1 when the LSI is started in user boot mode; otherwise, the initial value is 0. Writing to this bit is enabled only when this register is accessed in word size and H'3B is written to the KEY bits. |
|         |          |                  |        | 0: Selects the user MAT  |
|         |          |                  |        | 1: Selects the user boot MAT   |

Note: Write data is not retained.

# 25.3.6 FCU RAM Enable Register (FCURAME)

FCURAME enables or disables access to the FCU RAM area. In on-chip ROM disabled mode, FCURAME is read as H'00 and writing to it is ignored. FCURAME is initialized by a power-on reset or a transition to the hardware standby mode.



Note: \* Written data is not stored in these bits.

| Bit     | Bit Name | Initial<br>Value | R/W    | Description   |
|---------|----------|------------------|--------|---|
| 15 to 8 | KEY      | H'00             | R/(W)* | Key Code  |
|         |          |                  |        | These bits enable or disable FCRME bit modification. The data written to these bits are not stored.   |
| 7 to 1  | _        | All 0            | R      | Reserved  |
|         |          |                  |        | These bits are always read as 0. The write value should always be 0.  |
| 0       | FCRME    | 0                | R/W    | FCU RAM Enable  |
|         |          |                  |        | Enables or disables access to the FCU RAM. Writing to this bit is enabled only when this register is accessed in word size and H'C4 is written to the KEY bits. Before writing to the FCU RAM, clear FENTRYR to H'0000 to stop the FCU. |
|         |          |                  |        | 0: Disables access to FCU RAM   |
|         |          |                  |        | 1: Enables access to FCU RAM  |

Note: Write data is not retained.

# 25.3.7 Flash Status Register 0 (FSTATR0)

FSTATR0 indicates the FCU status. In on-chip ROM disabled mode, FSTATR0 is read as H'00. FRTATR0 is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of the FRESETR register is set to 1.

| Bit:           | 7    | 6           | 5          | 4          | 3          | 2 | 1          | 0          |
|----------------|------|-------------|------------|------------|------------|---|------------|------------|
|                | FRDY | ILG<br>LERR | ERS<br>ERR | PRG<br>ERR | SUS<br>RDY | _ | ERS<br>SPD | PRG<br>SPD |
| Initial value: |      | 0           | 0          | 0          | 0          | 0 | 0          | 0          |
| R/W:           | R    | R           | R          | R          | R          | R | R          | R          |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 7   | FRDY     | 1                | R   | Flash Ready   |
|     |          |                  |     | Indicates the processing state in the FCU.  |
|     |          |                  |     | <ol> <li>Programming or erasure processing,<br/>programming or erasure suspension processing,<br/>lock bit read 2 command processing, or EEPROM<br/>blank check is in progress (see section 26,<br/>EEPROM).</li> </ol> |
|     |          |                  |     | 1: None of the above is in progress.  |
| 6   | ILGLERR  | 0                | R   | Illegal Command Error   |
|     |          |                  |     | Indicates that the FCU has detected an illegal command or illegal ROM or EEPROM access. When this bit is 1, the FCU is in command-locked state (see section 25.8.3, Error Protection).                                  |
|     |          |                  |     | 0: The FCU has not detected any illegal command or illegal ROM/EEPROM access  |
|     |          |                  |     | <ol> <li>The FCU has detected an illegal command or<br/>illegal ROM/EEPROM access</li> </ol>  |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | The FCU has detected an illegal command.  |
|     |          |                  |     | <ul> <li>The FCU has detected an illegal ROM/EEPROM<br/>access (the ROMAE, EEPAE, EEPIFE, EEPRPE,<br/>or EEPWPE bit in FASTAT is 1).</li> </ul>   |
|     |          |                  |     | The FENTRYR setting is illegal.   |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | The FCU completes the status-clear command<br>processing while FASTAT is H'10.  |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 5   | ERSERR   | 0                | R   | Erasure Error  |
|     |          |                  |     | Indicates the result of ROM or EEPROM erasure by the FCU. When this bit is 1, the FCU is in command-locked state (see section 25.8.3, Error Protection).     |
|     |          |                  |     | Erasure processing has been completed successfully   |
|     |          |                  |     | 1: An error has occurred during erasure  |
|     |          |                  |     | [Setting conditions]   |
|     |          |                  |     | An error has occurred during erasure.  |
|     |          |                  |     | <ul> <li>A block erase command has been issued for the<br/>area protected by a lock bit.</li> </ul>  |
|     |          |                  |     | [Clearing condition]   |
|     |          |                  |     | The FCU completes the status-clear command   |
|     |          |                  |     | processing.  |
| 4   | PRGERR   | 0                | R   | Programming Error  |
|     |          |                  |     | Indicates the result of ROM or EEPROM programming by the FCU. When this bit is 1, the FCU is in command-locked state (see section 25.8.3, Error Protection). |
|     |          |                  |     | 0: Programming has been completed successfully   |
|     |          |                  |     | 1: An error has occurred during programming  |
|     |          |                  |     | [Setting conditions]   |
|     |          |                  |     | An error has occurred during programming.  |
|     |          |                  |     | <ul> <li>A programming command has been issued for the<br/>area protected by a lock bit.</li> </ul>  |
|     |          |                  |     | [Clearing condition]   |
|     |          |                  |     | The FCU completes the status-clear command processing.   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 3   | SUSRDY   | 0                | R   | Suspend Ready  |
|     |          |                  |     | Indicates whether the FCU is ready to accept a P/E suspend command.  |
|     |          |                  |     | 0: The FCU cannot accept a P/E suspend command   |
|     |          |                  |     | 1: The FCU can accept a P/E suspend command  |
|     |          |                  |     | [Setting condition]  |
|     |          |                  |     | <ul> <li>After initiating programming/erasure, the FCU has<br/>entered a state where it is ready to accept a P/E<br/>suspend command.</li> </ul> |
|     |          |                  |     | [Clearing conditions]  |
|     |          |                  |     | The FCU has accepted a P/E suspend command.  |
|     |          |                  |     | The FCU has entered a command-locked state   |
|     |          |                  |     | during programming or erasure.   |
| 2   | _        | 0                | R   | Reserved   |
|     |          |                  |     | This bit is always read as 0. Correct operation is not guaranteed if 1 is written to this bit.   |
| 1   | ERSSPD   | 0                | R   | Erasure-Suspended Status   |
|     |          |                  |     | Indicates that the FCU has entered an erasure suspension process or an erasure-suspended status (see section 25.6.4, Suspending Operation).      |
|     |          |                  |     | <ol> <li>The FCU is in a status other than the below-<br/>mentioned.</li> </ol>  |
|     |          |                  |     | 1: The FCU is in an erasure suspension process or an erasure-suspended status.   |
|     |          |                  |     | [Setting condition]  |
|     |          |                  |     | <ul> <li>The FCU has initiated an erasure suspend command.</li> </ul>  |
|     |          |                  |     | [Clearing condition]   |
|     |          |                  |     | The FCU has accepted a resume command.   |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 0   | PRGSPD   | 0                | R   | Programming-Suspended Status  |
|     |          |                  |     | Indicates that the FCU has entered a write suspension process or a write suspend status (see section 25.6.4, Suspending Operation). |
|     |          |                  |     | 0: The FCU is in a status other than the below-<br>mentioned.   |
|     |          |                  |     | <ol> <li>The FCU is in a write suspension process or a<br/>write-suspended status.</li> </ol>                                       |
|     |          |                  |     | [Setting condition]   |
|     |          |                  |     | The FCU has initiated a write suspend command.  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | The FCU has accepted a resume command.  |

# 25.3.8 Flash Status Register 1 (FSTATR1)

FSTATR1 indicates the FCU status. In on-chip ROM disabled mode, FSTATR1 is read as H'00. FSTATR1 is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of the FRESETR register is set to 1.

| Bit:           | 7          | 6 | 5 | 4           | 3 | 2 | 1          | 0          |
|----------------|------------|---|---|-------------|---|---|------------|------------|
|                | FCU<br>ERR | _ | _ | FLO<br>CKST | _ | _ | FRD<br>TCT | FRC<br>RCT |
| Initial value: | 0          | 0 | 0 | 0           | 0 | 0 | 0          | 0          |
| R/W:           | R          | R | R | R           | R | R | R          | R          |

|      |          | Initial |     |   |
|------|----------|---------|-----|---|
| Bit  | Bit Name | Value   | R/W | Description   |
| 7    | FCUERR   | 0       | R   | FCU Error   |
|      |          |         |     | Indicates an error has occurred during the CPU processing in the FCU.   |
|      |          |         |     | 0: No error has occurred during the CPU processing in the FCU   |
|      |          |         |     | <ol> <li>An error has occurred during the CPU processing<br/>in the FCU</li> </ol>  |
|      |          |         |     | [Clearing condition]  |
|      |          |         |     | • The FRESET bit in FRESETR is set to 1.  |
|      |          |         |     | When FCUERR is 1, set the FRESET bit to 1 to initialize the FCU, and then copy the FCU firmware again from the FCU firmware area to the FCU RAM area.   |
| 6, 5 | _        | All 0   | R   | Reserved  |
|      |          |         |     | These bits are always read as 0. The write value should always be 0.  |
| 4    | FLOCKST  | 0       | R   | Lock Bit Status   |
|      |          |         |     | Reflects the lock bit data read through lock bit read 2 command execution. When the FRDY bit becomes 1 after the lock bit read 2 command is issued, valid data is stored in this bit. This bit value is retained until the next lock bit read 2 command is completed. |
|      |          |         |     | 0: Protected state  |
|      |          |         |     | 1: Non-protected state  |
|      |          |         |     |   |

| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 3, 2 | _        | All 0            | R   | Reserved  |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 1    | FRDTCT   | 0                | R   | FCU RAM ECC 2-Bit Error Detection Monitoring Bit  |
|      |          |                  |     | Indicates that a 2-bit error has been detected when the FCU is reading RAM.   |
|      |          |                  |     | 0: No 2-bit error has been detected.  |
|      |          |                  |     | 1: A 2-bit error has been detected.   |
|      |          |                  |     | When FRDTCT is 1, set the FRESET bit to 1 to initialize the FCU, and then copy the FCU firmware again from the FCU firmware area to the FCU RAM area. |
| 0    | FRCRCT   | 0                | R   | FCU RAM ECC 1-Bit Error Correction Monitoring Bit   |
|      |          |                  |     | Indicates that a 1-bit error has been corrected when the FCU is reading RAM.  |
|      |          |                  |     | 0: No 1-bit error has been corrected.   |
|      |          |                  |     | 1: A 1-bit error has been corrected.  |
|      |          |                  |     | When FRCRCT is 1, set the FRESET bit to 1 to initialize the FCU, and then copy the FCU firmware again from the FCU firmware area to the FCU RAM area. |

## 25.3.9 FCU RAM ECC Error Control Register (FRAMECCR)

FRAMECCR enables or disables an FCU command lock request upon correction of an ECC 1-bit error or detection of an ECC 2-bit error when the FCU is reading the RAM. FRAMECCR enables or disables an FCU command lock request, but it does not control the FRDTCT and FRCRCT bits of the flash status register 1 (FSTATR1). In on-chip ROM disabled mode, FRAMECCR is read as H'00 and writing to it is ignored. FRAMECCR is initialized by a power-on reset or a transition to the hardware standby mode.

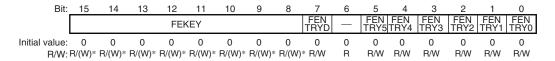
| Bit:           | 7 | 6 | 5 | 4 | 3 | 2 | 1          | 0          |
|----------------|---|---|---|---|---|---|------------|------------|
|                | _ | _ | _ | _ | _ |   | FRD<br>CLE | FRC<br>CLE |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 1          | 0          |
| R/W:           | R | R | R | R | R | R | R/W        | R/W        |

| Bit    | Bit Name | Initial<br>Value | R/W | Description   |
|--------|----------|------------------|-----|---|
| 7 to 2 | _        | All 0            | R   | Reserved  |
|        |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 1      | FRDCLE   | 1                | R/W | FCU Command Lock Enabling Bit upon FCU RAM 2-Bit Error Detection  |
|        |          |                  |     | Enables or disables an FCU command-lock request upon detection of an ECC 2-bit error when the FCU is reading RAM. If an ECC 2-bit error is detected while this bit is set to 1, the CMDLK bit of FASTAT is set to 1.  |
|        |          |                  |     | 0: Issues no FCU command-lock request upon detection of a 2-bit error (setting prohibited).   |
|        |          |                  |     | <ol> <li>Issues an FCU command-lock request upon<br/>detection of a 2-bit error.</li> </ol>   |
| 0      | FRCCLE   | 0                | R/W | FCU Command Lock Enabling Bit upon FCU RAM 1-Bit Error Correction   |
|        |          |                  |     | Enables or disables an FCU command-lock request upon correction of an ECC 1-bit error when the FCU is reading RAM. If an ECC 1-bit error is detected while this bit is set to 1, the CMDLK bit of FASTAT is set to 1. |
|        |          |                  |     | <ol> <li>Issues no FCU command-lock request upon<br/>detection of a 1-bit error.</li> </ol>   |
|        |          |                  |     | 1: Issues an FCU command-lock request upon detection of a 1-bit error.  |

### 25.3.10 Flash P/E Mode Entry Register (FENTRYR)

FENTRYR specifies the P/E mode for the ROM or EEPROM. To specify the P/E mode for the ROM or EEPROM so that the FCU can accept commands, set any one of the FENTRYD, FENTRY5, FENTRY4, FENTRY3, FENTRY2, FENTRY1, or FENTRY0 bit to 1. Note that if this register is set to other than H'0001, H'0002, H'0004, H'0008, H'0010, H'0020, and H'0080, the ILGLERR bit in the FSTATR0 register will be set and the FCU will enter command-locked state. In on-chip ROM disabled mode, FENTRYR is read as H'0000 and writing to it is ignored. FENTRYR can be initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

When transiting to ROM read mode by changing FENTRY5 to FENTRY0 bits in FENTRYR register from 1 to 0, write 0 to FENTRY5 to FENTRY0 bits and execute a minimum of 5 NOP instructions after performing a dummy read of FENTRYR register.



Notes: \* Written data is not stored in these bits.

|         |          | Initial |        |  |  |
|---------|----------|---------|--------|--|--|
| Bit     | Bit Name | Value   | R/W    | Description  |  |
| 15 to 8 | FEKEY    | All 0   | R/(W)* | Key Code   |  |
|         |          |         |        | These bits enable or disable rewriting of the FENTRYD and FENTRY5-FENTRY0 bits. Write data to these bits are not retained. |  |
| 7       | FENTRYD  | 0       | R/W    | EEPROM P/E Mode Entry Bit  |  |
|         |          |         |        | Refer to section 26, EEPROM.   |  |
| 6       | _        | 0       | R      | Reserved   |  |
|         |          |         |        | These bits are always read as 0. The write value should always be 0.   |  |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |  |
|-----|----------|------------------|-----|--|--|
| 5   | FENTRY5  | 0                | R/W | ROM P/E Mode Entry Bit 5   |  |
|     |          |                  |     | Specifies the P/E mode for the 0.25-Mbyte ROM (read addresses: H'00380000 to H'003BFFFF; program/erase addresses: H'80B80000 to H'80BBFFFF). |  |
|     |          |                  |     | 0: The 0.25-Mbyte ROM is in read mode.   |  |
|     |          |                  |     | 1: The 0.25-Mbyte ROM is in P/E mode.  |  |
|     |          |                  |     | Programming is enabled when the following conditions are all satisfied:  |  |
|     |          |                  |     | The LSI is in on-chip ROM enabled mode.  |  |
|     |          |                  |     | The FWE bit in FPMON is 1.   |  |
|     |          |                  |     | • The FRDY bit in FSTATR0 is 1.  |  |
|     |          |                  |     | H'AA is written to FEKEY in word access.   |  |
|     |          |                  |     | [Setting condition]  |  |
|     |          |                  |     | <ul> <li>1 is written to FENTRY5 while the write enabling<br/>conditions are satisfied and FENTRYR is H'0000.</li> </ul>                     |  |
|     |          |                  |     | [Clearing conditions]  |  |
|     |          |                  |     | <ul> <li>The FRDY bit in FSTATR0 becomes 1 and the<br/>FWE bit in FPMON becomes 0.</li> </ul>  |  |
|     |          |                  |     | This register is written to in byte access.  |  |
|     |          |                  |     | <ul> <li>A value other than H'AA is written to FEKEY in<br/>word access.</li> </ul>  |  |
|     |          |                  |     | <ul> <li>0 is written to FENTRY5 while the write enabling<br/>conditions are satisfied.</li> </ul>   |  |
|     |          |                  |     | <ul> <li>FENTRYR is written to while FENTRYR is not<br/>H'0000 and the write enabling conditions are<br/>satisfied.</li> </ul>               |  |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 4   | FENTRY4  | 0                | R/W | ROM P/E Mode Entry Bit 4   |
|     |          |                  |     | Specifies the P/E mode for the 0.25-Mbyte ROM (read addresses: H'00340000 to H'0037FFFF; program/erase addresses: H'80B40000 to H'80B7FFFF). |
|     |          |                  |     | 0: The 0.25-Mbyte ROM is in read mode.   |
|     |          |                  |     | 1: The 0.25-Mbyte ROM is in P/E mode.  |
|     |          |                  |     | Programming is enabled when the following conditions are all satisfied:  |
|     |          |                  |     | The LSI is in on-chip ROM enabled mode.  |
|     |          |                  |     | The FWE bit in FPMON is 1.   |
|     |          |                  |     | • The FRDY bit in FSTATR0 is 1.  |
|     |          |                  |     | <ul> <li>H'AA is written to FEKEY in word access.</li> </ul>   |
|     |          |                  |     | [Setting condition]  |
|     |          |                  |     | <ul> <li>1 is written to FENTRY4 while the write enabling<br/>conditions are satisfied and FENTRYR is H'0000.</li> </ul>                     |
|     |          |                  |     | [Clearing conditions]  |
|     |          |                  |     | <ul> <li>The FRDY bit in FSTATR0 becomes 1 and the<br/>FWE bit in FPMON becomes 0.</li> </ul>  |
|     |          |                  |     | This register is written to in byte access.  |
|     |          |                  |     | <ul> <li>A value other than H'AA is written to FEKEY in<br/>word access.</li> </ul>  |
|     |          |                  |     | <ul> <li>0 is written to FENTRY4 while the write enabling<br/>conditions are satisfied.</li> </ul>   |
|     |          |                  |     | <ul> <li>FENTRYR is written to while FENTRYR is not<br/>H'0000 and the write enabling conditions are<br/>satisfied.</li> </ul>               |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |  |
|-----|----------|------------------|-----|--|--|
| 3   | FENTRY3  | 0                | R/W | ROM P/E Mode Entry Bit 3   |  |
|     |          |                  |     | Specifies the P/E mode for the 0.25-Mbyte ROM (read addresses: H'00300000 to H'0033FFFF; program/erase addresses: H'80B00000 to H'80B3FFFF). |  |
|     |          |                  |     | 0: The 0.25-Mbyte ROM is in read mode.   |  |
|     |          |                  |     | 1: The 0.25-Mbyte ROM is in P/E mode.  |  |
|     |          |                  |     | Programming is enabled when the following conditions are all satisfied:  |  |
|     |          |                  |     | The LSI is in on-chip ROM enabled mode.  |  |
|     |          |                  |     | The FWE bit in FPMON is 1.   |  |
|     |          |                  |     | • The FRDY bit in FSTATR0 is 1.  |  |
|     |          |                  |     | H'AA is written to FEKEY in word access.   |  |
|     |          |                  |     | [Setting condition]  |  |
|     |          |                  |     | <ul> <li>1 is written to FENTRY3 while the write enabling<br/>conditions are satisfied and FENTRYR is H'0000.</li> </ul>                     |  |
|     |          |                  |     | [Clearing conditions]  |  |
|     |          |                  |     | <ul> <li>The FRDY bit in FSTATR0 becomes 1 and the<br/>FWE bit in FPMON becomes 0.</li> </ul>  |  |
|     |          |                  |     | This register is written to in byte access.  |  |
|     |          |                  |     | <ul> <li>A value other than H'AA is written to FEKEY in<br/>word access.</li> </ul>  |  |
|     |          |                  |     | 0 is written to FENTRY3 while the write enabling conditions are satisfied.   |  |
|     |          |                  |     | <ul> <li>FENTRYR is written to while FENTRYR is not<br/>H'0000 and the write enabling conditions are<br/>satisfied.</li> </ul>               |  |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 2   | FENTRY2  | 0                | R/W | ROM P/E Mode Entry Bit 2   |
|     |          |                  |     | These bits specify the P/E mode for the 1-Mbyte ROM (read addresses: H'00200000 to H'002FFFFF; program/erase addresses: H'80A00000 to H'80AFFFFF). |
|     |          |                  |     | 0: the 1-Mbyte ROM is in read mode   |
|     |          |                  |     | 1: the 1-Mbyte ROM is in P/E mode  |
|     |          |                  |     | Programming is enabled when the following conditions are all satisfied:  |
|     |          |                  |     | The LSI is in on-chip ROM enabled mode.  |
|     |          |                  |     | The FWE bit in FPMON is 1.   |
|     |          |                  |     | The FRDY bit in FSTATR0 is 1.  |
|     |          |                  |     | H'AA is written to FEKEY in word access.   |
|     |          |                  |     | [Setting condition]  |
|     |          |                  |     | <ul> <li>1 is written to FENTRY2 while the write enabling<br/>conditions are satisfied and FENTRYR is H'0000.</li> </ul>                           |
|     |          |                  |     | [Clearing conditions]  |
|     |          |                  |     | The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0.  |
|     |          |                  |     | This register is written to in byte access.  |
|     |          |                  |     | <ul> <li>A value other than H'AA is written to FEKEY in<br/>word access.</li> </ul>  |
|     |          |                  |     | <ul> <li>0 is written to FENTRY2 while the write enabling<br/>conditions are satisfied.</li> </ul>   |
|     |          |                  |     | <ul> <li>FENTRYR is written to while FENTRYR is not<br/>H'0000 and the write enabling conditions are<br/>satisfied.</li> </ul>                     |

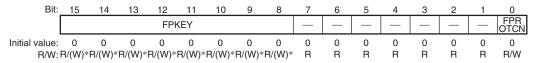
| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 1   | FENTRY1  | 0                | R/W | ROM P/E Mode Entry Bit 1   |
|     |          |                  |     | These bits specify the P/E mode for the 1-Mbyte ROM (read addresses: H'00100000 to H'001FFFFF; program/erase addresses: H'80900000 to H'809FFFFF).   |
|     |          |                  |     | 0: the 1-Mbyte ROM is in read mode   |
|     |          |                  |     | 1: the 1-Mbyte ROM is in P/E mode  |
|     |          |                  |     | Programming is enabled when the following conditions are all satisfied:  |
|     |          |                  |     | The LSI is in on-chip ROM enabled mode.  |
|     |          |                  |     | The FWE bit in FPMON is 1.   |
|     |          |                  |     | The FRDY bit in FSTATR0 is 1.  |
|     |          |                  |     | H'AA is written to FEKEY in word access.   |
|     |          |                  |     | [Setting condition]  |
|     |          |                  |     | 1 is written to FENTRY while the write enabling conditions are satisfied and FENTRYR is H'0000.  TO A STATE OF THE PROPERTY WITH A STATE OF THE PROPERTY |
|     |          |                  |     | [Clearing conditions]  |
|     |          |                  |     | <ul> <li>The FRDY bit in FSTATR0 becomes 1 and the<br/>FWE bit in FPMON becomes 0.</li> </ul>  |
|     |          |                  |     | This register is written to in byte access.  |
|     |          |                  |     | <ul> <li>A value other than H'AA is written to FEKEY in<br/>word access.</li> </ul>  |
|     |          |                  |     | <ul> <li>0 is written to FENTRY while the write enabling<br/>conditions are satisfied.</li> </ul>  |
|     |          |                  |     | <ul> <li>FENTRYR is written to while FENTRYR is not<br/>H'0000 and the write enabling conditions are<br/>satisfied.</li> </ul>   |

| Bit | Bit Name | Initial<br>Value | R/W | Description  |
|-----|----------|------------------|-----|--|
| 0   | FENTRY0  | 0                | R/W | ROM P/E Mode Entry Bit 0   |
|     |          |                  |     | These bits specify the P/E mode for the 1-Mbyte ROM (read addresses: H'00000000 to H'000FFFFF; program/erase addresses: H'80800000 to H'808FFFFF). |
|     |          |                  |     | 0: the 1-Mbyte ROM is in read mode   |
|     |          |                  |     | 1: the 1-Mbyte ROM is in P/E mode  |
|     |          |                  |     | Programming is enabled when the following conditions are all satisfied:  |
|     |          |                  |     | The LSI is in on-chip ROM enabled mode.  |
|     |          |                  |     | The FWE bit in FPMON is 1.   |
|     |          |                  |     | The FRDY bit in FSTATR0 is 1.  |
|     |          |                  |     | H'AA is written to FEKEY in word access.   |
|     |          |                  |     | [Setting condition]  |
|     |          |                  |     | • 1 is written to FENTRY while the write enabling conditions are satisfied and FENTRYR is H'0000.  |
|     |          |                  |     | [Clearing conditions]  |
|     |          |                  |     | <ul> <li>The FRDY bit in FSTATR0 becomes 1 and the<br/>FWE bit in FPMON becomes 0.</li> </ul>  |
|     |          |                  |     | This register is written to in byte access.  |
|     |          |                  |     | <ul> <li>A value other than H'AA is written to FEKEY in<br/>word access.</li> </ul>  |
|     |          |                  |     | <ul> <li>0 is written to FENTRY while the write enabling<br/>conditions are satisfied.</li> </ul>  |
|     |          |                  |     | <ul> <li>FENTRYR is written to while FENTRYR is not<br/>H'0000 and the write enabling conditions are<br/>satisfied.</li> </ul>                     |

Note: \* Write data is not retained.

## 25.3.11 Flash Protect Register (FPROTR)

FPROTR enables or disables the protection function through the lock bits against programming and erasure. In on-chip ROM disabled mode, FPROTR is read as H'0000 and writing to it is ignored. FPROTR is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.



Note: \* Written data is not stored in these bits.

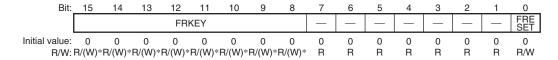
1-- 141-1

| D:4     | D'A M    | Initial | D 04/  | Book total  |
|---------|----------|---------|--------|---|
| Bit     | Bit Name | Value   | R/W    | Description   |
| 15 to 8 | FPKEY    | H'00    | R/(W)* | Key Code  |
|         |          |         |        | These bits enable or disable FPROTCN bit modification. The data written to these bits are not stored.   |
| 7 to 1  | _        | All 0   | R      | Reserved  |
|         |          |         |        | These bits are always read as 0. The write value should always be 0.  |
| 0       | FPROTCN  | 0       | R/W    | Lock Bit Protect Cancel   |
|         |          |         |        | Enables or disables protection through the lock bits against programming and erasure.   |
|         |          |         |        | 0: Enables protection through the lock bits   |
|         |          |         |        | 1: Disables protection through the lock bits  |
|         |          |         |        | [Setting condition]   |
|         |          |         |        | <ul> <li>H'55 is written to FPKEY and 1 is written to<br/>FPROTCN in word access while the FENTRYR<br/>register value is not H'0000.</li> </ul> |
|         |          |         |        | [Clearing conditions]   |
|         |          |         |        | <ul> <li>This register is written to in byte access.</li> </ul>   |
|         |          |         |        | <ul> <li>A value other than H'55 is written to FPKEY in<br/>word access.</li> </ul>   |
|         |          |         |        | H'55 is written to FPKEY and 0 is written to  |
|         |          |         |        | FPROTCN in word access.   |
|         |          |         |        | The FENTRYR register value is H'0000.   |

Note: \* Write data is not retained.

## 25.3.12 Flash Reset Register (FRESETR)

FRESETR is used for the initialization of FCU. In on-chip ROM disabled mode, FRESETR is read as H'0000 and writing to it is ignored. FRESETR is initialized by a power-on reset or a transition to the hardware standby mode.



Note: \* Written data is not stored in these bits.

|         |          | Initial |        |  |  |
|---------|----------|---------|--------|--|--|
| Bit     | Bit Name | Value   | R/W    | Description  |  |
| 15 to 8 | FRKEY    | H'00    | R/(W)* | Key Code   |  |
|         |          |         |        | These bits enable or disable FRESET bit modification. The data written to these bits are not stored. |  |
| 7 to 1  | _        | All 0   | R      | Reserved   |  |
|         |          |         |        | These bits are always read as 0. The write value should always be 0.                                 |  |
| 0       | FRESET   | 0       | R/W    | Flash Reset  |  |
|         |          |         |        | ,  |  |
|         |          |         |        | 0: Issue no reset to the FCU.  |  |
|         |          |         |        | 1: Issues a reset to the FCU.  |  |

Note: \* Write data is not retained.

# 25.3.13 FCU Command Register (FCMDR)

FCMDR stores the commands that the FCU has accepted. In on-chip ROM disabled mode, FCMDR is read as H'0000 and writing to it is ignored. FCMDR is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.



| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 15 to 8 | CMDR     | H'FF             | R   | Command Register   |
|         |          |                  |     | These bits store the latest command accepted by the FCU.   |
| 7 to 0  | PCMDR    | H'FF             | R   | Precommand Register  |
|         |          |                  |     | These bits store the previous command accepted by the FCU. |

Table 25.3 shows the states of FCMDR after acceptance of the various commands. For details on the blank check, see section 26.6, User Mode, User Program Mode, and User Boot Mode.

Table 25.3 FCMDR Status after a Command is Accepted

| Command   | CMDR | PCMDR            |
|---|------|------------------|
| Normal mode transition                          | H'FF | Previous command |
| Status read mode transition                     | H'70 | Previous command |
| Lock bit read mode transition (lock bit read 1) | H'71 | Previous command |
| Program   | H'E8 | Previous command |
| Block erase                                     | H'D0 | H'20             |
| P/E suspend                                     | H'B0 | Previous command |
| P/E resume                                      | H'D0 | Previous command |
| Status register clear                           | H'50 | Previous command |
| Lock bit read 2, blank check                    | H'D0 | H'71             |
| Lock bit program                                | H'D0 | H'77             |

# 25.3.14 FCU Processing Switch Register (FCPSR)

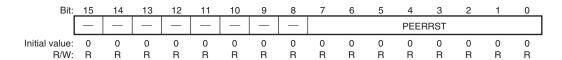
FCPSR selects a function to make the FCU suspend erasure. In on-chip ROM disabled mode, FCPSR is read as H'0000 and writing to it is ignored. FCPSR is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.



| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 15 to 1 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 0       | ESUSPMD  | 0                | R/W | Erasure-Suspended Mode   |
|         |          |                  |     | Selects the erasure-suspended mode to be entered when a P/E suspend command is issued while the FCU is erasing the ROM or EEPROM (see section 25.6.4, Suspending Operation). |
|         |          |                  |     | 0: Suspension-priority mode  |
|         |          |                  |     | 1: Erasure-priority mode   |

# Flash P/E Status Register (FPESTAT)

FPESTAT indicates the result of programming/erasure of the ROM/EEPROM. In on-chip ROM disabled mode, FPESTAT is read as H'0000 and writing to it is ignored. FPESTAT is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.



| Bit     | Bit Name | Initial<br>Value | R/W | Description   |  |
|---------|----------|------------------|-----|---|--|
| 15 to 8 | _        | All 0            | R   | Reserved  |  |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0.  |  |
| 7 to 0  | PEERRST  | H'00             | R   | P/E Error Status  |  |
|         |          |                  |     | Indicates the source of an error that occurs during programming/erasure. This bit value is only valid if the PRGERR or ERSERR bit value in FSTATR0 is 1; otherwise the bit retains the value to indicate the source of an error that previously occurred. |  |
|         |          |                  |     | H'01: A write attempt made to an area protected by the lock bits  |  |
|         |          |                  |     | H'02: A write error caused by other source than the above   |  |
|         |          |                  |     | H'11: An erase attempt made to an area protected by the lock bits   |  |
|         |          |                  |     | H'12: An erase error caused by other source than the above  |  |
|         |          |                  |     | Other than above: Reserved  |  |

## 25.4 Overview of ROM-Related Modes

Figure 25.4 shows the ROM-related mode transition in this LSI. For the relationship between the LSI operating modes and the MD4 to MD0 and FWE pin settings, refer to section 3, Operating Modes.

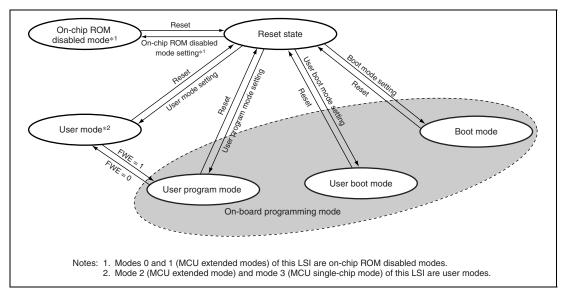


Figure 25.4 ROM-Related Mode Transition

- The ROM cannot be read, programmed, or erased in on-chip ROM disabled mode.
- The ROM can be read but cannot be programmed or erased in user mode.
- The ROM can be read, programmed, and erased on the board in user program mode, user boot mode, and boot mode.

Table 25.4 compares programming- and erasure-related items for the boot mode, user program mode, and user boot mode.

**Table 25.4 Comparison of Programming Modes** 

| Item                             | <b>Boot Mode</b>              | User Program Mode          | User Boot Mode                |
|----------------------------------|-------------------------------|----------------------------|-------------------------------|
| Programming/erasure enabled MAT  | User MAT and user boot MAT    | User MAT                   | User MAT                      |
| Programming/erasure control      | Host                          | FCU                        | FCU                           |
| Programming data transfer        | From host via SCI             | From any device via<br>RAM | From any device via<br>RAM    |
| Reset-start MAT                  | Embedded program stored MAT   | User MAT                   | User boot MAT*                |
| Transition to MCU operating mode | Mode setting change and reset | FWE setting change         | Mode setting change and reset |

Note: \* After the LSI is started in the embedded program stored MAT and the embedded program built in the product is executed, execution starts from the location indicated by the reset vector of the user boot MAT.

- The user boot MAT can be programmed or erased only in boot mode.
- In user boot mode, a boot operation with a desired interface can be implemented through mode pin settings different from those in user program mode.
- In boot mode or user boot mode, the embedded program built in the product uses H'FFF88000 to H'FFF8FFFF in the on-chip RAM. Therefore, once the RAM is disabled via the RAM enable control register (RAMEN) and a reset is issued, the data stored in the corresponding area in the RAM prior to the reset is no longer retained in the RAM after booting is initiated in boot mode or user boot mode (see section 28, RAM).

### 25.5 Boot Mode

### 25.5.1 System Configuration

To program or erase the user MAT and user boot MAT in boot mode, send control commands and programming data from the host. The on-chip SCI of this LSI is used in asynchronous mode for communications between the host and this LSI. The tool for sending control commands and programming data must be prepared in the host. When this LSI is started in boot mode, the program in the embedded program stored MAT is executed. This program automatically adjusts the SCI bit rate and performs communications between the host and this LSI by means of the control command method.

Figure 25.5 shows the system configuration in boot mode. The NMI and  $\overline{IRQ7}$  to  $\overline{IRQ0}$  interrupts are ignored in this mode, but these pins must be fixed to non-active state. Note that the AUD cannot be used in this mode.

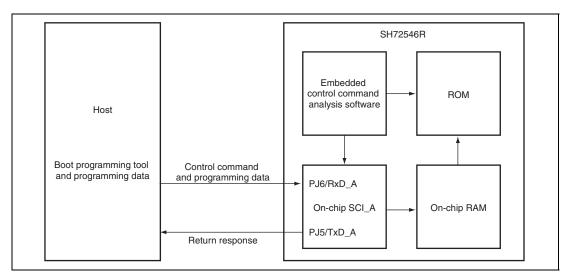


Figure 25.5 System Configuration in Boot Mode

### 25.5.2 State Transition in Boot Mode

Figure 25.6 shows the state transition in boot mode.

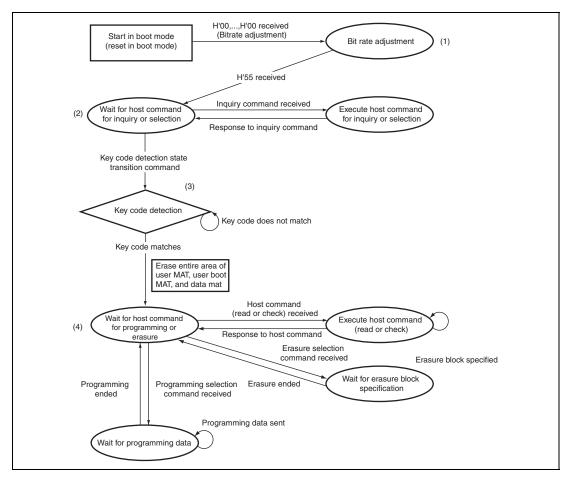


Figure 25.6 State Transition in Boot Mode

# (1) Bit Rate Adjustment

After this LSI is started in boot mode, it automatically adjusts the bit rate for communications between the host and SCI\_A. After automatic adjustment of the bit rate, the LSI sends H'00 to the host. After the LSI has successfully received H'55 sent from the host, the LSI waits for a host command for inquiry or selection. For details on bit rate adjustment, see section 25.5.3, Automatic Adjustment of Bit Rate.

### (2) Waiting for Host Command for Inquiry or Selection

In this state, the host inquires regarding MAT information (such as the size, configuration, and start address) and the supported functions, and selects the device, clock mode, and bit rate. Upon reception of a key code detection state transition command sent from the host, this LSI enters the key code detection state. For details of inquiry/selection host commands, see section 25.5.4, Inquiry/Selection Host Command Wait State.

### (3) Key Code Detection

In this state, this LSI checks whether the key code sent by the host matches the one stored in this LSI. When the code matches, this LSI erases the entire area of each of the user MAT, user boot MAT, and data MAT, and enters the programming/erasure command wait state. When the key code of this LSI is in the initial state (meaning nothing has been written), approvals to any key code can be performed.

### (4) Waiting for Host Command for Programming or Erasure

In this state, this LSI performs programming or erasure according to the command sent from the host. The LSI enters programming data wait state, erasure block specification wait state, or command (read or check) processing state depending on the received command.

Upon reception of a programming selection command, the LSI waits for programming data. After the programming selection command, send the programming start address and programming data from the host. Specifying H'FFFFFFF as the programming start address terminates programming processing and the LSI makes a transition from the programming data wait state to programming/erasure command wait state.

Upon reception of an erasure selection command, the LSI waits for erasure block specification. After the erasure selection command, send the erasure block number from the host. Specifying H'FF as the erasure block number terminates erasure processing and the LSI makes a transition from the erasure block specification wait state to programming/erasure command wait state. As the entire area of each of the user MAT, user boot MAT, and EEPROM data MAT is erased before the LSI enters programming/erasure command wait state after it is started in boot mode, erasure processing is not needed except for the case when the data programmed in boot mode should be erased without resetting the LSI.

In addition to programming and erasing commands, many other host commands are provided for use in programming/erasure command wait state; these include commands for checksum, blank check (erasure check)\*, memory read, and status inquiry. For details on these host commands, see section 25.5.5, Programming/Erasing Host Command Wait State.

Note: \* Blank check function checks the erase state of the area where erase has ended. The function is disabled when programming/erasing was suspended (e.g., reset input, power-supply interruption).

### 25.5.3 Automatic Adjustment of Bit Rate

When this LSI is started in boot mode, it measures the low-level (H'00) period of the data that is continuously sent from the host in asynchronous SCI communications. During this measurement, set the SCI transmit/receive format to 8-bit data, 1 stop bit, and no parity, and set the bit rate to 9,600 bps or 19,200 bps. This LSI calculates the bit rate of the host SCI by means of the measured low-level period, and then sends H'00 to the host after completing the bit rate adjustment. When the host has received H'00 successfully, it must send H'55 to this LSI. If the host has failed to receive H'00, restart this LSI in boot mode to calculate and adjust the bit rate again. When this LSI has received H'55, it returns H'E6 to the host, or when it has failed to receive H'55, it returns H'EF.

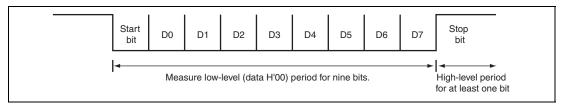


Figure 25.7 SCI Transmit/Receive Format for Automatic Adjustment of Bit Rate

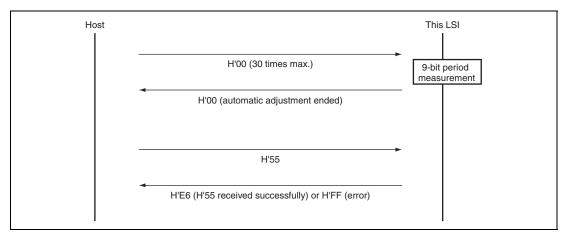


Figure 25.8 Communication Sequence between Host and this LSI

The bit rate may not be adjusted correctly depending on the bit rate of the host SCI or the peripheral clock frequency of this LSI. Satisfy the SCI communications condition as shown in table 25.5.

Table 25.5 Condition for Automatic Adjustment of Bit Rate

| Host SCI Bit Rate | Peripheral Clock Frequency of this LSI |
|-------------------|--|
| 9,600 bps         | 16 MHz to 20 MHz, 32 MHz to 40 MHz     |
| 19,200 bps        |  |

## 25.5.4 Inquiry/Selection Host Command Wait State

Table 25.6 shows the host commands available in inquiry/selection host command wait state. The boot program status inquiry command can also be used in programming/erasure host command wait state. The other commands can only be used in inquiry/selection host command wait state.

Table 25.6 Inquiry/Selection Host Commands

| Supported device inquiry  Inquires regarding the device codes and the product codes for the embedded programs  Selects a device code  Clock mode inquiry  Inquires regarding the clock mode  Clock mode selection  Selects a clock mode  Multiplication ratio inquiry  Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication ratios  Operating frequency inquiry  Inquires regarding the number of clock types and the maximum and minimum operating frequencies  User boot MAT information inquiry  Inquires regarding the number of user boot MATs and the start and end addresses  User MAT information inquiry  Inquires regarding the number of user MATs and the start and end addresses  Erasure block information inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the size of programming data  Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition  Enters the key code detection state  Key code check command  Sends the key code  Boot program status inquiry  Inquires regarding the state of this LSI | Host Command Name                   | Function  |
|---|-------------------------------------|---|
| Clock mode inquiry  Inquires regarding the clock mode  Selects a clock mode  Multiplication ratio inquiry  Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication/division ratios  Operating frequency inquiry  Inquires regarding the number of clock types and the maximum and minimum operating frequencies  User boot MAT information inquiry  Inquires regarding the number of user boot MATs and the start and end addresses  User MAT information inquiry  Inquires regarding the number of user MATs and the start and end addresses  Erasure block information inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the size of programming data  New bit rate selection  Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition  Enters the key code detection state  Sends the key code  | Supported device inquiry            |   |
| Clock mode selection  Selects a clock mode  Multiplication ratio inquiry  Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication/division ratios  Operating frequency inquiry  Inquires regarding the number of clock types and the maximum and minimum operating frequencies  User boot MAT information inquiry  Inquires regarding the number of user boot MATs and the start and end addresses  User MAT information inquiry  Inquires regarding the number of user MATs and the start and end addresses  Erasure block information inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the size of programming data  New bit rate selection  Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition  Enters the key code detection state  Key code check command  Sends the key code   | Device selection                    | Selects a device code                                     |
| Multiplication ratio inquiry  Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication ratios  Operating frequency inquiry  Inquires regarding the number of clock types and the maximum and minimum operating frequencies  User boot MAT information inquiry  Inquires regarding the number of user boot MATs and the start and end addresses  User MAT information inquiry  Inquires regarding the number of user MATs and the start and end addresses  Erasure block information inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the size of programming data  New bit rate selection  Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition  Enters the key code detection state  Sends the key code  | Clock mode inquiry                  | Inquires regarding the clock mode                         |
| of multiplication/division ratios, and the multiplication /division ratios  Operating frequency inquiry  Inquires regarding the number of clock types and the maximum and minimum operating frequencies  User boot MAT information inquiry  Inquires regarding the number of user boot MATs and the start and end addresses  User MAT information inquiry  Inquires regarding the number of user MATs and the start and end addresses  Erasure block information inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the size of programming data  New bit rate selection  Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition  Enters the key code detection state  Key code check command   | Clock mode selection                | Selects a clock mode                                      |
| The start and minimum operating frequencies  User boot MAT information inquiry  Inquires regarding the number of user boot MATs and the start and end addresses  User MAT information inquiry  Inquires regarding the number of user MATs and the start and end addresses  Erasure block information inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the size of programming data  Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition  Enters the key code detection state  Key code check command  Sends the key code   | Multiplication ratio inquiry        | of multiplication/division ratios, and the multiplication |
| Start and end addresses  User MAT information inquiry  Inquires regarding the number of user MATs and the start and end addresses  Erasure block information inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the size of programming data  Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition  Enters the key code detection state  Key code check command  Sends the key code  | Operating frequency inquiry         |   |
| and end addresses  Erasure block information inquiry  Inquires regarding the number of blocks and the start and end addresses  Programming size inquiry  Inquires regarding the size of programming data  Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition  Enters the key code detection state  Key code check command  Sends the key code   | User boot MAT information inquiry   |   |
| end addresses  Programming size inquiry Inquires regarding the size of programming data  New bit rate selection Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition Enters the key code detection state  Key code check command Sends the key code   | User MAT information inquiry        |   |
| New bit rate selection  Modifies the bit rate of SCI communications between the host and this LSI  Key code detection state transition  Enters the key code detection state  Key code check command  Sends the key code   | Erasure block information inquiry   |   |
| host and this LSI  Key code detection state transition  Enters the key code detection state  Key code check command  Sends the key code   | Programming size inquiry            | Inquires regarding the size of programming data           |
| Key code check command Sends the key code   | New bit rate selection              |   |
| •   | Key code detection state transition | Enters the key code detection state                       |
| Boot program status inquiry Inquires regarding the state of this LSI  | Key code check command              | Sends the key code  |
|   | Boot program status inquiry         | Inquires regarding the state of this LSI                  |

If the host has sent an undefined command, this LSI returns a response indicating a command error in the format shown below. The command field holds the first byte of the undefined command sent from the host.

| Error response  | H'80 | Command |  |
|-----------------|------|---------|--|
| Elloi lespolise | 1160 | Command |  |

In inquiry/selection host command wait state, send selection commands from the host in the order of device selection, clock mode selection, and new bit rate selection to set up this LSI according to the responses to inquiry commands. Note that the supported device inquiry and clock mode inquiry commands are the only inquiry commands that can be sent before the clock mode selection command; other inquiry commands must not be issued before the clock mode selection command.

If commands are issued in an incorrect order, this LSI returns a response indicating a command error. Figure 25.9 shows an example of the procedure to use inquiry/selection host commands.

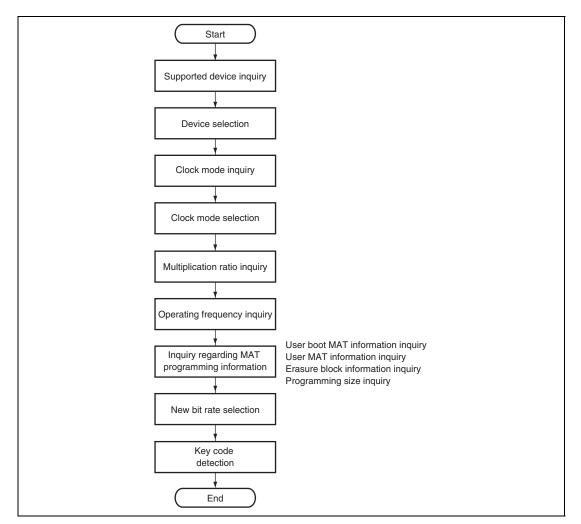


Figure 25.9 Example of Procedure to Use Inquiry/Selection Host Commands

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

# (1) Supported Device Inquiry

In response to a supported device inquiry command sent from the host, this LSI returns the information concerning the devices supported by the embedded program for boot mode. If the supported device inquiry command comes after the host has selected a device, this LSI only returns the information concerning the selected device.

| Command | H'2 |
|---------|-----|
| Command | H'2 |

| H'30            | Size | Device count |    |              |
|-----------------|------|--------------|----|--------------|
| Character count |      | Device co    | de | Product code |
| Character count |      | Device co    | de | Product code |
| :               |      | :            |    | :            |
| Character count |      | Device co    | de | Product code |
| SUM             |      |              |    |              |

[Legend]

Size (1 byte): Total number of bytes in the device count, character count, device code, and

product code fields

Device count (1 byte): Number of device types supported by the embedded program for boot

mode

Character count (1 byte): Number of characters included in the device code and product code

fields

Device code (4 bytes): ASCII code for the product name of the chip Product code (n bytes): ASCII code for the name of the supported device

SUM (1 byte): Checksum

Page 1260 of 1824

#### **Device Selection (2)**

In response to a device selection command sent from the command, this LSI checks if the selected device is supported. When the selected device is supported, this LSI specifies this device as the device for use and returns a response (H'06). If the selected device is not supported or the sent command is illegal, this LSI returns an error response (H'90).

Even when H'01 has been returned as the number of supported devices in response to a supported device inquiry command, issue a device selection command to specify the device code that has been returned as the result of the inquiry.

| Command        | H'10 | Size  | Device code | SUM |
|----------------|------|-------|-------------|-----|
|                |      |       |             |     |
| Response       | H'06 |       |             |     |
|                |      |       |             |     |
| Error response | H'90 | Error |             |     |

[Legend]

Size (1 byte): Number of characters in the device code field (fixed at four)

Device code (4 bytes): ASCII code for the product name of the chip (one of the device codes

returned in response to the supported device inquiry command)

SUM (1 byte): Checksum Error (1 byte): Error code

H'11: Checksum error (illegal command)

H'21: Incorrect device code error

#### **Clock Mode Inquiry (3)**

In response to a clock mode inquiry command sent from the host, this LSI returns the supported clock modes. If the clock mode inquiry command comes after the host has selected a clock mode, this LSI only returns the information concerning the selected clock mode.

Command H'21

Response

H'31 Size Mode Mode Mode . . . SUM

[Legend]

Size (1 byte): Total number of bytes in the mode count and mode fields

Mode (1 byte): Supported clock mode (for example, H'01 indicates clock mode 1)

SUM (1 byte): Checksum

### (4) Clock Mode Selection

In response to a clock mode selection command sent from the host, this LSI checks if the selected clock mode is supported. When the selected mode is supported, this LSI specifies this clock mode for use and returns a response (H'06). If the selected mode is not supported or the sent command is illegal, this LSI returns an error response (H'91).

Be sure to issue a clock mode selection command only after issuing a device selection command. Even when H'00 or H'01 has been returned as the number of supported clock modes in response to a clock mode inquiry command, issue a clock mode selection command to specify the clock mode that has been returned as the result of the inquiry.

| Command        | H'11 | Size  | Mode | SUM |
|----------------|------|-------|------|-----|
| Response       | H'06 |       |      |     |
| Error response | H'91 | Error |      |     |

[Legend]

Size (1 byte): Number of characters in the mode field (fixed at 1)

Mode (1 byte): Clock mode (one of the clock modes returned in response to the clock mode

inquiry command)

SUM (1 byte): Checksum Error (1 byte): Error code

H'11: Checksum error (illegal command)

H'22: Incorrect clock mode error

## (5) Multiplication Ratio Inquiry

In response to a multiplication ratio inquiry command sent from the host, this LSI returns the clock types, the number of multiplication/division ratios, and the multiplication division ratios supported.

Command

H'22

Response

| H'32                       | Size                 | Clock type count     |                          |
|----------------------------|----------------------|----------------------|--------------------------|
| Multiplication ratio count | Multiplication ratio | Multiplication ratio | <br>Multiplication ratio |
| Multiplication ratio count | Multiplication ratio | Multiplication ratio | <br>Multiplication ratio |
| :                          | :                    | :                    | <br>:                    |
| Multiplication ratio count | Multiplication ratio | Multiplication ratio | <br>Multiplication ratio |
| SUM                        |                      |                      |                          |

[Legend]

Size (1 byte): Total number of bytes in the clock type count, multiplication ratio count, and

multiplication ratio fields

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock

types; that is, an internal clock and a peripheral clock)

Multiplication ratio count (1 byte): Number of supported multiplication/division ratios (for

example, H'03 indicates that three multiplication ratios are

supported for the internal clock (x4, x6, and x8))

Multiplication ratio (1 byte): A positive value indicates a multiplication ratio (for example,

H'04 = 4 = multiplication by 4)

A negative value indicates a division ratio (for example,

H'FE = -2 = division by 2

SUM (1 byte): Checksum

# (6) Operating Clock Frequency Inquiry

In response to an operating clock frequency inquiry command sent from the host, this LSI returns the minimum and maximum frequencies for each clock.

Command

H'23

Response

| H'33    | Size      | Clock type count  |   |  |
|---------|-----------|-------------------|---|--|
| Minimum | frequency | Maximum frequency |   |  |
| Minimum | frequency | Maximum frequency |   |  |
| :       | •         |                   | : |  |
| Minimum | frequency | Maximum frequency |   |  |
| SUM     |           |                   |   |  |

[Legend]

Size (1 byte): Total number of bytes in the clock type count, minimum frequency, and

maximum frequency fields

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock

types; that is, an internal clock and a peripheral clock)

Minimum frequency (2 bytes): Minimum value of the operating frequency (for example,

H'07D0 indicates 20.00 MHz).

This value should be calculated by multiplying the frequency

value (MHz) to two decimal places by 100.

Maximum frequency (2 bytes): Maximum value of the operating frequency represented in the

same format as the minimum frequency

# (7) User Boot MAT Information Inquiry

In response to a user boot MAT information inquiry command sent from the host, this LSI returns the number of user boot MATs and their addresses.

Response

H'34

Size

MAT count

MAT start address

MAT end address

MAT end address

MAT end address

MAT end address

:

MAT start address

MAT end address

MAT end address

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end

address fields

SUM

MAT count (1 byte): Number of user boot MATs (consecutive areas are counted as one

MAT)

MAT start address (4 bytes): Start address of a user boot MAT MAT end address (4 bytes): End address of a user boot MAT

# (8) User MAT Information Inquiry

In response to a user MAT information inquiry command sent from the host, this LSI returns the number of user MATs and their addresses.

| Command | H'25 |
|---------|------|
|         |      |

Response

| H'35            | Size              | MAT count |  |  |  |  |  |  |
|-----------------|-------------------|-----------|--|--|--|--|--|--|
|                 | MAT start address |           |  |  |  |  |  |  |
|                 | MAT end address   |           |  |  |  |  |  |  |
|                 | MAT start address |           |  |  |  |  |  |  |
|                 | MAT end address   |           |  |  |  |  |  |  |
|                 |                   | :         |  |  |  |  |  |  |
|                 | MAT start address |           |  |  |  |  |  |  |
| MAT end address |                   |           |  |  |  |  |  |  |
| SUM             |                   |           |  |  |  |  |  |  |

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end

address fields

MAT count (1 byte): Number of user MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of a user MAT MAT end address (4 bytes): End address of a user MAT

# (9) Erasure Block Information Inquiry

In response to an erasure block information inquiry command sent from the host, this LSI returns the number of erasure blocks in the user MAT and their addresses.

Command

H'26

Response

| H'36              | Size                | Block count |  |  |  |  |  |
|-------------------|---------------------|-------------|--|--|--|--|--|
|                   | Block start address |             |  |  |  |  |  |
|                   | Block end address   |             |  |  |  |  |  |
|                   | Block start address |             |  |  |  |  |  |
|                   | Block end address   |             |  |  |  |  |  |
|                   | :                   |             |  |  |  |  |  |
|                   | Block start address |             |  |  |  |  |  |
| Block end address |                     |             |  |  |  |  |  |
| SUM               |                     |             |  |  |  |  |  |

[Legend]

Size (2 bytes): Total number of bytes in the block count, block start address, and block end

address fields

Block count (1 byte): Number of erasure blocks in the user MAT Block start address (4 bytes): Start address of an erasure block

Block end address (4 bytes): End address of an erasure block

# (10) Programming Size Inquiry

In response to a programming size inquiry command sent from the host, this LSI returns the programming size.

Command

H'27

Response

H'37 Size Programming size SUM

[Legend]

Size (1 byte): Number of characters included in the programming size field (fixed at two)

Programming size (2 bytes): Programming unit (bytes)

#### (11) New Bit Rate Selection

In response to a new bit rate selection command sent from the host, this LSI checks if the on-chip SCI can be set to the selected new bit rate. When the SCI can be set to the new bit rate, this LSI returns a response (H'06) and sets the SCI to the new bit rate. If the SCI cannot be set to the new bit rate or the sent command is illegal, this LSI returns an error response (H'BF). Upon reception of response H'06, the host waits for a one-bit period in the previous bit rate with which the new bit rate selection command has been sent, and then sets the host bit rate to the new one. After that, the host sends confirmation data (H'06) in the new bit rate, and this LSI returns a response (H'06) to the confirmation data.

Be sure to issue a new bit rate selection command only after a clock mode selection command.

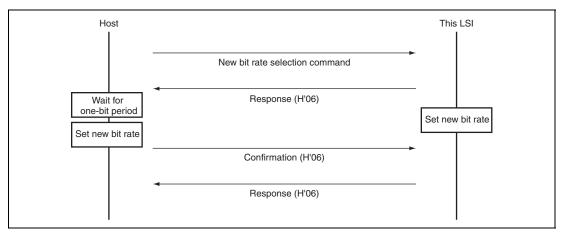


Figure 25.10 New Bit Rate Selection Sequence

| Command        | H'3F             | Size                   | Bit ra                 | ate | Input frequency |
|----------------|------------------|------------------------|------------------------|-----|-----------------|
|                | Clock type count | Multiplication ratio 1 | Multiplication ratio 2 |     |                 |
|                | SUM              |                        |                        |     |                 |
|                |                  | <b>.</b>               |                        |     |                 |
| Response       | H'06             |                        |                        |     |                 |
|                |                  | •                      |                        |     |                 |
| Error response | H'BF             | Error                  |                        |     |                 |
|                |                  |                        | •                      |     |                 |
| Confirmation   | H'06             |                        |                        |     |                 |
|                |                  | <u>I</u>               |                        |     |                 |
| Response       | H'06             |                        |                        |     |                 |
|                |                  | I                      |                        |     |                 |
|                |                  | •                      |                        |     |                 |

[Legend]

Size (1 byte): Total number of bytes in the bit rate, input frequency, clock type count, and

multiplication ratio fields

Bit rate (2 bytes): New bit rate (for example, H'00C0 indicates 19200 bps)

1/100 of the new bit rate value should be specified.

Input frequency (2 bytes): Clock frequency input to this LSI (for example, H'07D0 indicates

20.00 MHz)

This value should be calculated by multiplying the input

frequency value to two decimal places by 100.

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock

types; that is, an internal clock and a peripheral clock)

Multiplication ratio 1 (1 byte): Multiplication/division ratio of the input frequency to obtain

the internal clock

A positive value indicates a multiplication ratio (for example,

H'04 = 4 = multiplication by 4

A negative value indicates a division ratio (for example, HFE

= -2 = division by 2

Multiplication 2 (1 byte): Multiplication/division ratio of the input frequency to obtain the

peripheral clock

This value is represented in the same format as multiplication ratio 1

Error: Error code

H'11: Checksum error

H'24: Bit rate selection error H'25: Input frequency error H'26: Multiplication ratio error H'27: Operating frequency error

#### Bit rate selection error

A bit rate selection error occurs when the bit rate selected through a new bit rate selection command cannot be set for the SCI of this LSI within an error of 4%. The bit rate error can be obtained by the following equation from the bit rate (B) selected through a new bit rate selection command, the input frequency (fEX), multiplication ratio 2 (MP $\phi$ ), the SCBRR1A setting (N) in SCI\_A, and the CKS[1:0] bit value (n) in SCSMR1A.

Error (%) = 
$$\left\{ \frac{f_{EX} \times M_{P_{\phi}} \times 10^{6}}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$

### • Input frequency error

An input frequency error occurs when the input frequency specified through a new bit rate selection command is outside the range from the minimum to maximum input frequencies for the clock mode selected through a clock mode selection command.

# • Multiplication ratio error

A multiplication ratio error occurs when the multiplication ratio specified through a new bit rate selection command does not match the clock mode selected through a clock mode selection command. To check the selectable multiplication ratios, issue a multiplication ratio inquiry command.

# • Operating frequency error

An operating frequency error occurs when this LSI cannot operate at the operating frequencies selected through a new bit rate selection command. This LSI calculates the operating frequencies from the input frequency and multiplication ratios specified through a new bit rate selection command and checks if each calculated frequency is within the range from the minimum to maximum frequencies for the respective clock. To check the minimum and maximum operating frequencies for each clock, issue an operating clock frequency inquiry command.

### (12) Key Code Detection State Transition

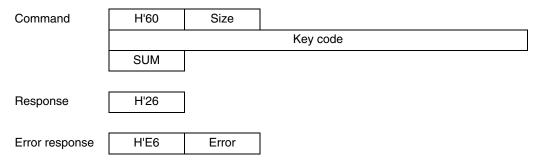
When receiving a key code detection state transition command, this LSI enters the key code detection state.

Command H'40 Response H'16

### (13) Key Code Check Command

When receiving a key code check command, this LSI compares the key code sent from the host and the one in the MAT. When the key codes match, this LSI returns a response (H'26), erases the entire area of each of the user MAT, user boot MAT, and data MAT, then enters the programming/erasure state. When the key codes do not match, this LSI returns an error response (H'E0).

When the key code of this LSI is in the initial state (meaning nothing has been written), this LSI returns a response (H'26) to any key code, erases the entire area of each of the user MAT, user boot MAT, and data MAT, then enters the programming/erasure state.



[Legend]

Size (1 byte): Number of characters in the key code (fixed at 16)

Key code (16 bytes): Key code for boot mode approval; pad higher-order bits that are not

required with H'FF.

SUM (1 byte): Checksum (summary from command to SUM, set as H'00)

Error code Error (1 byte): H'11: Checksum error

H'61: Incorrect key code error

H'51: Erasure cannot be done due to an erasure error

# (14) Boot Program Status Inquiry

In response to a boot program status inquiry command sent from the host, this LSI returns its current status. The boot program status inquiry command can be issued in both inquiry/selection host command wait state and programming/erasure host command wait state.

Command H'4F

H'5F Size Status Error SUM Response

[Legend]

Size (1 byte): Total number of bytes in the status and error fields (fixed at two)

Current status in this LSI (see table 25.7) Status (1 byte): Error (1 byte): Error status in this LSI (see table 25.8)

Table 25.7 Status Code

| Code | Description   |
|------|---|
| H'11 | Waiting for device selection  |
| H'12 | Waiting for clock mode selection  |
| H'13 | Waiting for new bit rate selection  |
| H'1F | Waiting for transition to key code detection state (new bit rate has been selected) |
| H'3F | Waiting for a programming/erasure host command                                      |
| H'4F | Waiting for reception of programming data   |
| H'5F | Waiting for erasure block selection   |

# Table 25.8 Error Code

| Code | Description                            |
|------|--|
| H'00 | No error                               |
| H'11 | Checksum error                         |
| H'21 | Incorrect device code error            |
| H'22 | Incorrect clock mode error             |
| H'24 | Bit rate selection error               |
| H'25 | Input frequency error                  |
| H'26 | Multiplication ratio error             |
| H'27 | Operating frequency error              |
| H'29 | Block number error                     |
| H'2A | Address error                          |
| H'2B | Data size error                        |
| H'51 | Erasure error                          |
| H'52 | Incomplete erasure error               |
| H'53 | Programming error                      |
| H'61 | Incorrect key code error               |
| H'80 | Command error                          |
| H'FF | Bit rate adjustment verification error |

### 25.5.5 Programming/Erasing Host Command Wait State

Table 25.9 shows the host commands available in programming/erasure host command wait state.

Table 25.9 Programming/Erasure Host Commands

| Host Command Name                   | Function   |
|-------------------------------------|--|
| User boot MAT programming selection | Selects the program for user boot MAT programming    |
| User MAT programming selection      | Selects the program for user MAT programming         |
| 256-byte programming                | Programs 256 bytes of data                           |
| Erasure selection                   | Selects the erasure program                          |
| Block erasure                       | Erases block data                                    |
| Memory read                         | Reads data from memory                               |
| User boot MAT checksum              | Performs checksum verification for the user boot MAT |
| User MAT checksum                   | Performs checksum verification for the user MAT      |
| User boot MAT blank check           | Checks whether the user boot MAT is blank            |
| User MAT blank check                | Checks whether the user MAT is blank                 |
| Read lock bit status                | Reads from the lock bit                              |
| Lock bit program                    | Writes to the lock bit                               |
| Lock bit enabled                    | Enables the lock bit protect                         |
| Lock bit disable                    | Disables the lock bit protect                        |
| Boot program status inquiry         | Inquires regarding the state of this LSI             |

If the host has sent an undefined command, this LSI returns a response indicating a command error. For the format of this response, see section 25.5.4, Inquiry/Selection Host Command Wait State.

To program the ROM, issue a programming selection command (user boot MAT programming selection or user MAT programming selection command) and then a 256-byte programming command from the host. Upon reception of a programming selection command, this LSI enters programming data wait state (see section 25.5.2, State Transition in Boot Mode). In response to a 256-byte programming command sent from the host in this state, this LSI starts programming the ROM. When the host sends a 256-byte programming command specifying H'FFFFFFF as the programming start address, this LSI detects it as the end of programming and enters programming/erasure host command wait state.

To erase the ROM, issue an erasure selection command and then a block erasure command from the host. Upon reception of an erasure selection command, this LSI enters erasure block selection wait state (see section 25.5.2, State Transition in Boot Mode). In response to a block erasure command sent from the host in this state, this LSI erases the specified block in the ROM. When the host sends a block erasure command specifying H'FF as the block number, this LSI detects it as the end of erasure and enters programming/erasure host command wait state.

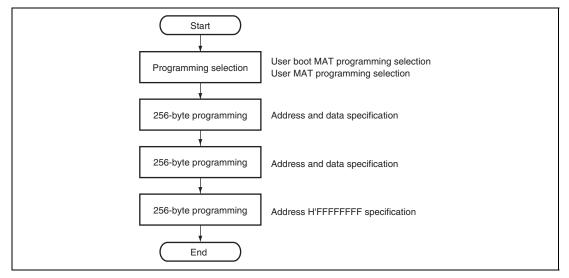


Figure 25.11 Procedure for ROM Programming in Boot Mode

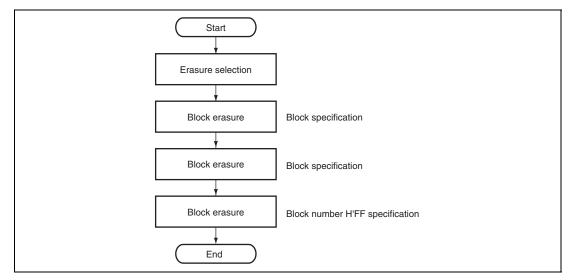


Figure 25.12 Procedure for ROM Erasure in Boot Mode

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

# (1) User Boot MAT Programming Selection

In response to a user boot MAT programming selection command sent from the host, this LSI selects the program for user boot MAT programming and waits for programming data.

Command H'42

Response H'06

# (2) User MAT Programming Selection

In response to a user MAT programming selection command sent from the host, this LSI selects the program for user MAT programming and waits for programming data.

Command H'43

Response H'06

### (3) 256-Byte Programming

In response to a 256-byte programming command sent from the host, this LSI programs the ROM. After completing ROM programming successfully, this LSI returns a response (H'06). If an error has occurred during ROM programming, this LSI returns an error response (H'D0).

| Command | H'50 | Programming Address |  |      |   |  |  |
|---------|------|---------------------|--|------|---|--|--|
|         | Data | Data                |  | Data |   |  |  |
|         | SUM  |                     |  |      | • |  |  |
|         |      | •                   |  |      |   |  |  |

Response H'06

Error response H'D0 Error

[Legend]

Programming address (4 bytes): Target address of programming

To program the ROM, a 256-byte boundary address should be

specified.

To terminate programming, H'FFFFFFF should be specified.

Data (256 bytes): Programming data

H'FF should be specified for the bytes that do not need to be programmed. When terminating programming, no data needs to be specified (only the

programming address and SUM should be sent in that order).

SUM (1 byte): Checksum Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error (the specified address is not in the target MAT) H'53: Programming cannot be done due to a programming error

### (4) Erasure Selection

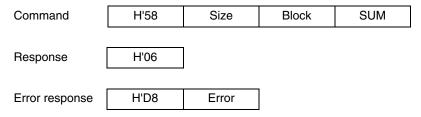
In response to an erasure selection command sent from the host, this LSI selects the erasure program and waits for erasure block specification.

Command H'48

Response H'06

#### (5) Block Erasure

In response to a block erasure command sent from the host, this LSI erases the ROM. After completing ROM erasure successfully, this LSI returns a response (H'06). If an error has occurred during ROM erasure, this LSI returns an error response (H'D8).



[Legend]

Size (1 byte): Number of bytes in the block specification field (fixed at 1)

Block (1 byte): Block number whose data is to be erased

To terminate erasure, H'FF should be specified.

SUM (1 byte): Checksum Error (1 byte): Error code

H'11: Checksum error

H'29: Block number error (an incorrect block number is specified)

H'51: Erasure cannot be done due to an erasure error

#### (6)**Memory Read**

In response to a memory read command sent from the host, this LSI reads data from the ROM. After completing ROM reading, this LSI returns the data stored in the address specified by the memory read command. If this LSI has failed to read the ROM, this LSI returns an error response (H'D2).

| Command        | H'52             | Size         | Area | Rea  | s |  |
|----------------|------------------|--------------|------|------|---|--|
|                | Reading size SUN |              |      |      |   |  |
|                |                  |              |      |      |   |  |
| Response       | H'52             | Reading size |      |      |   |  |
|                | Data             | Data         |      | Data |   |  |
|                | SUM              |              |      |      |   |  |
|                |                  |              |      |      |   |  |
| Error response | H'D2             | Error        | ]    |      |   |  |

[Legend]

Size (1 byte): Total number of bytes in the area, read start address, and reading size fields

Area (1 byte): Target MAT to be read

H'00: User boot MAT

H'01: User MAT

Read start address (4 bytes): Start address of the area to be read

Reading size (4 bytes): Size of data to be read (bytes)

Checksum SUM (1 byte):

Data read from the ROM Data (1 byte):

Error (1 byte): Error code

> H'11: Checksum error H'2A: Address error

- The value specified for area selection is neither H'00 nor H'01.
- The specified read start address is outside the selected MAT.

H'2B: Data size error

- H'00 is specified for the reading size.
- The reading size is larger than the MAT.
- The end address calculated from the read start address and the reading size is outside the selected MAT.

#### (7) User Boot MAT Checksum

In response to a user boot MAT checksum command sent from the host, this LSI sums the user boot MAT data in byte units and returns the result (checksum).

Command

H'4A

Response

H'5A Size

MAT checksum

SUM

[Legend]

Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4)

MAT checksum (4 bytes): Checksum of the user boot MAT data

SUM (1 byte): Checksum (for the response data)

#### (8) User MAT Checksum

In response to a user MAT checksum command sent from the host, this LSI sums the user MAT data in byte units and returns the result (checksum).

Command

H'4B

Response

H'5B Size

MAT checksum

SUM

[Legend]

Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4)

MAT checksum (4 bytes): Checksum of the user MAT data

The user MAT also stores the key code for boot mode/debugging function authentication. Note that the checksum includes this key

code value.

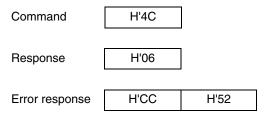
The indicated checksum is for 4 Mbytes. Since the user MAT of this LSI is 3.75 Mbytes, the unused 0.25 Mbytes are added as

H'FF in the indicated checksum.

SUM (1 byte): Checksum (for the response data)

#### (9) User Boot MAT Blank Check

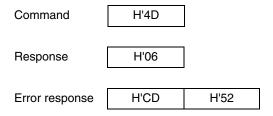
In response to a user boot MAT blank check command sent from the host, this LSI checks whether the user boot MAT is completely erased. When the user boot MAT is completely erased, this LSI returns a response (H'06). If the user boot MAT has an unerased area, this LSI returns an error response (sends H'CC and H'52 in that order).



No verification function is provided to check program/erase state of the area where the data is undefined by suspend of program/erase (e.g., reset input, power-supply interruption). Therefore, if the undefined area should be used again, make sure to completely erase data before usage.

#### (10) User MAT Blank Check

In response to a user MAT blank check command sent from the host, this LSI checks whether the user MAT is completely erased. When the user MAT is completely erased, this LSI returns a response (H'06). If the user MAT has an unerased area, this LSI returns an error response (sends H'CD and H'52 in that order).



No verification function is provided to check program/erase state of the area where the data is undefined by suspend of program/erase (e.g., reset input, power-supply interruption). Therefore, if the undefined area should be used again, make sure to completely erase data before usage.

#### (11) Read Lock Bit Status

In response to a read lock bit status command sent from the host, this LSI reads data from the lock bit. After completing the lock bit reading, this LSI returns the data stored in the address specified by the read lock bit status command. If this LSI has failed to read the lock bit, this LSI returns an error response (H'F1).

| Command        | H'71   | Size  | Area | Medium<br>address | Upper<br>address | SUM |
|----------------|--------|-------|------|-------------------|------------------|-----|
| Response       | Status |       |      |                   |                  |     |
| Error response | H'F1   | Error |      |                   |                  |     |

[Legend]

Size (1 byte): Total number of bytes in the area, medium address, and upper address (fixed at 3

in this LSI)

Area (1 byte): Target MAT to be read

H'00: User boot MAT

H'01: User MAT

Medium address (1 byte): Medium address at the end of the specified address (8 to 15 bits)

Upper address (1 byte): Upper address at the end of the specified address (16 to 23 bits)

SUM (1 byte): Checksum

Status (1 byte): Bit 6 locked at "0"

Bit 6 unlocked at "1"

Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error (the specified address is not in the target MAT)

### (12) Lock Bit Program

In response to a lock bit program command sent from the host, this LSI writes to a lock bit and locks the specified block. After completing the lock bit blocking, this LSI returns a response (H'06). If this LSI has failed to lock, this LSI returns an error response (H'F7).

| Command        | H'77 | Size  | Area | Medium<br>address | Upper<br>address | SUM |
|----------------|------|-------|------|-------------------|------------------|-----|
| Response       | H'06 |       |      |                   |                  |     |
| Error response | H'F7 | Error | 7    |                   |                  |     |

[Legend]

Size (1 byte): Total number of bytes in the area, medium address, and upper address (fixed at 3

in this LSI)

Area (1 byte): Target MAT to be locked

H'00: User boot MAT

H'01: User MAT

Medium address (1 byte): Medium address at the end of the specified address (8 to 15 bits)

Upper address (1 byte): Upper address at the end of the specified address (16 to 23 bits)

SUM (1 byte): Checksum Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error (the specified address is not in the target MAT)

H'53: Locking cannot be done due to a programming error

#### (13) Lock Bit Enable

In response to a lock bit enable command sent from the host, this LSI enables a lock bit.

Command H'7A

Response H'06

# (14) Lock Bit Disable

In response to a lock bit enable command sent from the host, this LSI disables a lock bit.

Command H'75

Response H'06

### (15) Boot Program Status Inquiry

For details, refer to section 25.5.4, Inquiry/Selection Host Command Wait State.

# 25.6 User Program Mode

#### 25.6.1 FCU Command List

To program or erase the user MAT in user program mode, issue FCU commands to the FCU. Table 25.10 is a list of FCU commands for ROM programming and erasure.

Table 25.10 FCU Command List (ROM-Related Commands)

| Command   | Function   |  |  |  |  |  |  |  |  |
|---|--|--|--|--|--|--|--|--|--|
| Normal mode transition                          | Moves to the normal mode (see section 25.6.2, Conditions for FCU Command Acceptance)                               |  |  |  |  |  |  |  |  |
| Status read mode transition                     | Moves to the status read mode (see section 25.6.2,<br>Conditions for FCU Command Acceptance)                       |  |  |  |  |  |  |  |  |
| Lock bit read mode transition (lock bit read 1) | Moves to the lock bit read mode (see section 25.6.2,<br>Conditions for FCU Command Acceptance)                     |  |  |  |  |  |  |  |  |
| Program   | Programs ROM (in 256-byte units)   |  |  |  |  |  |  |  |  |
| Block erase                                     | Erases ROM (in block units; erasing the lock bit)  |  |  |  |  |  |  |  |  |
| P/E suspend                                     | Suspends programming or erasure  |  |  |  |  |  |  |  |  |
| P/E resume                                      | Resumes programming or erasure   |  |  |  |  |  |  |  |  |
| Status register clear                           | Clears the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and cancels the command-locked state                        |  |  |  |  |  |  |  |  |
| Lock bit read 2                                 | Reads the lock bit of a specified erasure block (updates the FLOCKST bit in FSTATR1 to reflect the lock bit state) |  |  |  |  |  |  |  |  |
| Lock bit program                                | Writes to the lock bit of a specified erasure block  |  |  |  |  |  |  |  |  |

FCU commands other than the lock bit read 2 program and lock bit program are also used for EEPROM programming and erasure. When a lock bit read 2 command is issued to the EEPROM, an EEPROM blank check is executed. When a lock bit program command is issued to the EEPROM, it is detected as an illegal command and generates an error (see section 26, EEPROM).

To issue a command to the FCU, write to a ROM program/erase address through the P bus. Table 25.11 shows the FCU command format. Performing P-bus write access as shown in table 25.11 under specified conditions starts each command processing in the FCU. For the conditions for FCU command acceptance, refer to section 25.6.2, Conditions for FCU Command Acceptance. For details of each FCU command, refer to section 25.6.3, FCU Command Usage.

When H'71 is sent in the first cycle of an FCU command while the FRDMD bit is 0 (memory area read mode), the FCU accepts the lock bit read mode transition command (lock bit read 1). When a ROM program/erase address is read through the P bus after transition to the lock bit read mode, the FCU copies the lock bit of the erasure block corresponding to the accessed address into all bits in the read data. When H'71 is sent in the first cycle of the FCU command while the FRDMD bit is 1 (register read mode), the FCU waits for the second-cycle data (H'D0) of the lock bit read 2 command. When a ROM program/erase address is written to through the P bus in this state, the FCU copies the lock bit of the erasure block corresponding to the accessed address into the FLOCKST bit in FSTATR1.

There are two suspending modes to be initiated by the P/E suspend command; the suspension-priority mode and erasure-priority mode. For details of each mode, refer to section 25.6.4, Suspending Operation.

#### **Table 25.11 FCU Command Format**

|   | Number<br>of Bus<br>Cycles | First Cycle |      | Second Cycle |      | Third C | ycle | Fourth to 130th<br>Cycles |      | 131st Cycle |      |
|---|----------------------------|-------------|------|--------------|------|---------|------|---------------------------|------|-------------|------|
| Command   |                            | Address     | Data | Address      | Data | Address | Data | Address                   | Data | Address     | Data |
| Normal mode transition                          | 1                          | RA          | H'FF | _            | _    | _       | _    | _                         | _    | _           | _    |
| Status read mode transition                     | 1                          | RA          | H'70 | _            | _    | _       | _    | _                         | _    | =           | _    |
| Lock bit read mode transition (lock bit read 1) | 1                          | RA          | H'71 | _            | _    | _       | _    | _                         | _    | _           | _    |
| Program   | 131                        | RA          | H'E8 | RA           | H'80 | WA      | WD1  | RA                        | WDn  | RA          | H'D0 |
| Block erase                                     | 2                          | RA          | H'20 | ВА           | H'D0 | _       | _    | _                         | _    | _           | _    |
| P/E suspend                                     | 1                          | RA          | H'B0 | _            | _    | _       | _    | _                         | _    | _           | _    |
| P/E resume                                      | 1                          | RA          | H'D0 | _            | _    | _       | _    | _                         | _    | _           | _    |
| Status register clear                           | 1                          | RA          | H'50 | _            | _    | _       | _    | _                         | _    | _           | _    |
| Lock bit read 2                                 | 2                          | RA          | H'71 | ВА           | H'D0 | _       | _    | _                         | _    | _           | _    |
| Lock bit program                                | 2                          | RA          | H'77 | ВА           | H'D0 | _       | _    | <u> </u>                  | _    | _           | _    |

### [Legend]

RA: ROM program/erase address

When FENTRY0 is 1: An address in the range from H'80800000 to H'808FFFF When FENTRY1 is 1: An address in the range from H'80900000 to H'809FFFF When FENTRY2 is 1: An address in the range from H'80800000 to H'808FFFF When FENTRY3 is 1: An address in the range from H'80800000 to H'8087FFF When FENTRY4 is 1: An address in the range from H'80840000 to H'8087FFF When FENTRY5 is 1: An address in the range from H'80880000 to H'808FFFF

WA: ROM program address

Start address of 256-byte programming data

BA: ROM erasure block address

An address in the target erasure block (specified by the ROM program/erase address)

WDn: n-th word of programming data (n = 1 to 128)

# 25.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 25.13 is an FCU mode transition diagram.

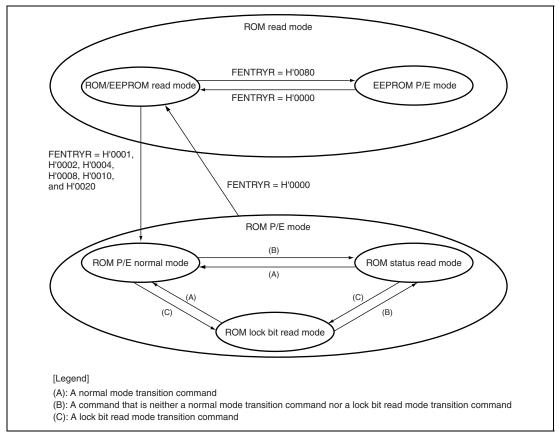


Figure 25.13 FCU Mode Transition Diagram (ROM-Related Modes)

#### (1) ROM Read Mode

#### ROM/EEPROM read mode

The ROM and EEPROM can be read through the ROM cache and peripheral bus, respectively, at a high speed. The FCU does not accept commands. The FCU enters this mode when the FENTRY5 to FENTRY0 bits in FENTRYR are set to 000000 and the FENTRYD bit to 0 in FENTRYR.

When transiting to ROM read mode by changing FENTRY5 to FENTRY0 bits in FENTRYR register from 1 to 0, write 0 to FENTRY5 to FENTRY0 bits and execute a minimum of 5 NOP instructions after performing a dummy read of FENTRYR register.

#### • EEPROM P/E mode

The ROM can be read through the ROM cache at a high speed. The FCU accepts commands for EEPROM, but does not accept commands for ROM. The FCU enters this mode when the FENTRY5 to FENTRY0 bits are set to 000000 and the FENTRYD bit to 1. For details of the EEPROM P/E mode, refer to section 26.6.2, Conditions for FCU Command Acceptance.

### (2) ROM P/E Mode

#### • ROM P/E normal mode

The FCU enters this mode when the FENTRYD bit is set to 0 and any one of the bits, FENTRY5 to FENTRY0, is set to 1 in ROM read mode, or when a normal mode transition command is accepted in ROM P/E mode. Table 25.12 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. If an address in the range from H'80B80000 to H'80B8FFFF is read through the P-bus while the FENTRY5 bit is set to 1, an address in the range from H'80B40000 to H'80B7FFFF is read through the P-bus while the FENTRY4 bit is set to 1, an address in the range from H'80B00000 to H'80B3FFFF is read through the P-bus while the FENTRY3 bit is set to 1, an address in the range from H'80A00000 to H'80AFFFFF is read through the P-bus while the FENTRY1 bit is set to 1, or an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY1 bit is set to 1, or an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY0 bit is set to 1, a ROM access error occurs and the FCU enters the command-locked state (see section 25.8.3, Error Protection).

#### ROM status read mode

The FCU enters this mode when the FCU accepts a command that is neither a normal mode transition command nor a lock bit read mode transition command in ROM P/E mode. The ROM status read mode includes the state in which the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 25.12 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. If an address in the range from H'80B80000 to H'80B8FFFF is read through the P-bus while the FENTRY5 bit is set to 1, an address in the range from H'80B40000 to H'80B7FFFF is read through the P-bus while the FENTRY4 bit is set to 1, an address in the range from H'80B00000 to H'80B3FFFF is read through the P-bus while the FENTRY3 bit is set to 1, an address in the range from H'80A00000 to H'80AFFFFF is read through the P-bus while the FENTRY2 bit is set to 1, an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY1 bit is set to 1, or an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY0 bit is set to 1, the FSTATR0 value is read.

#### ROM lock bit read mode

The FCU enters this mode when the FCU accepts a lock bit read mode transition command in ROM P/E mode. Table 25.12 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. The FENTRYR value is the same as that in ROM P/E normal mode. If an address in the range from H'80B80000 to H'80BBFFFF is read through the P-bus while the FENTRY5 bit is set to 1, an address in the range from H'80B40000 to H'80B7FFFF is read through the P-bus while the FENTRY4 bit is set to 1, an address in the range from H'80B00000 to H'80B3FFFF is read through the P-bus while the FENTRY3 bit is set to 1, an address in the range from H'80A00000 to H'80AFFFFF is read through the P-bus while the FENTRY2 bit is set to 1, an address in the range from H'80900000 to H'809FFFFF is read through the P-bus while the FENTRY1 bit is set to 1, or an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY0 bit is set to 1, the lock bit value of the target erasure block is returned through all bits in the read data.

Table 25.12 shows the acceptable commands in each FCU mode/state. When a command that cannot be accepted is issued, the FCU enters the command-locked state (see section 25.8.3, Error Protection).

To make sure that the FCU accepts a command, enter the mode in which the FCU can accept the target command, check the FRDY, ILGLERR, ERSERR, and PRGERR bit values in FSTATR0, and the FCUERR, FRDTCT, and FRCRCT bit values in FSTATR1, and then issue the target FCU command. The CMDLK bit in FASTAT holds a value obtained by logical ORing the ILGLERR, ERSERR, and PRGERR bit values in FSTATR0 and the FCUERR, FRDTCT, and FRCRCT bit values in the FSTATR1. Therefore the FCU's error occurrence state can be checked by reading the

CMDLK bit. In table 25.12, the CMDLK bit is used as the bit to indicate the error occurrence state. The FRDY bit of FSTATR0 is 0 during the programming/erasure, programming/erasure suspension, and lock bit read 2 processes. While the FRDY bit is 0, the P/E suspend command can be accepted only when the SUSRDY bit in FSTATR0 is 1.

Table 25.12 includes 0 and 1 in single cells of the ERSSPD, PRGSPD, and FRDY bit rows for the sake of simplification. The ERSSPD bits 1 and 0 indicate the erasure suspension and programming suspension processes, respectively. The PRGSPD bits 1 and 0 indicate the programming suspension and erasure suspension processes, respectively. The FRDY bit value can be either 1 or 0, which is a value held by the bit prior to a transition to the command lock state.

Table 25.12 FCU Modes/States and Acceptable Commands

|   | P/E                       | Nor<br>Iode       |             | Status Read Mode                  |   |  |                               |                           |                   |                           |                           |             |                           | Lock Bit<br>Read Mode |             |  |
|---|---------------------------|-------------------|-------------|-----------------------------------|---|--|-------------------------------|---------------------------|-------------------|---------------------------|---------------------------|-------------|---------------------------|-----------------------|-------------|--|
| Item  | Programming-<br>Suspended | Erasure-Suspended | Other State | Programming/Erasure<br>Processing | Programming<br>Processing during<br>Erasure-Suspended | Programming/Erasure<br>Suspension Processing | Lock Bit Read 2<br>Processing | Programming-<br>Suspended | Erasure-Suspended | Command-Locked (FRDY = 0) | Command-Locked (FRDY = 1) | Other State | Programming-<br>Suspended | Erasure-Suspended     | Other State |  |
| FRDY bit in FSTATR0                             | 1                         | 1                 | 1           | 0                                 | 0   | 0  | 0                             | 1                         | 1                 | 0                         | 1                         | 1           | 1                         | 1                     | 1           |  |
| SUSRDY bit in FSTATR0                           | 0                         | 0                 | 0           | 1                                 | 0   | 0  | 0                             | 0                         | 0                 | 0                         | 0                         | 0           | 0                         | 0                     | 0           |  |
| ERSSPD bit in FSTATR0                           | 0                         | 1                 | 0           | 0                                 | 1   | 0/1  | 0/1                           | 0                         | 1                 | 0/1                       | 0/1                       | 0           | 0                         | 1                     | 0           |  |
| PRGSPD bit in FSTATR0                           | 1                         | 0                 | 0           | 0                                 | 0   | 0/1  | 0/1                           | 1                         | 0                 | 0/1                       | 0/1                       | 0           | 1                         | 0                     | 0           |  |
| CMDLK bit in FASTAT                             | 0                         | 0                 | 0           | 0                                 | 0   | 0  | 0                             | 0                         | 0                 | 1                         | 1                         | 0           | 0                         | 0                     | 0           |  |
| Normal mode transition                          | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                             | Α                         | Α                 | ×                         | ×                         | Α           | Α                         | Α                     | Α           |  |
| Status read mode transition                     | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                             | Α                         | Α                 | ×                         | ×                         | Α           | Α                         | Α                     | Α           |  |
| Lock bit read mode transition (lock bit read 1) | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                             | Α                         | Α                 | ×                         | ×                         | Α           | Α                         | Α                     | Α           |  |
| Program   | ×                         | *                 | Α           | ×                                 | ×   | ×  | ×                             | ×                         | *                 | ×                         | ×                         | Α           | ×                         | *                     | Α           |  |
| Block erase                                     | ×                         | ×                 | Α           | ×                                 | ×   | ×  | ×                             | ×                         | ×                 | ×                         | ×                         | Α           | ×                         | ×                     | Α           |  |
| P/E suspend                                     | ×                         | ×                 | ×           | Α                                 | ×   | ×  | ×                             | ×                         | ×                 | ×                         | ×                         | ×           | ×                         | ×                     | ×           |  |
| P/E resume                                      | Α                         | Α                 | ×           | ×                                 | ×   | ×  | ×                             | Α                         | Α                 | ×                         | ×                         | ×           | Α                         | Α                     | ×           |  |
| Status register clear                           | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                             | Α                         | Α                 | ×                         | Α                         | Α           | Α                         | Α                     | Α           |  |
| Lock bit read 2                                 | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                             | Α                         | Α                 | ×                         | ×                         | Α           | Α                         | Α                     | Α           |  |
| Lock bit program                                | ×                         | *                 | Α           | ×                                 | ×   | ×  | ×                             | ×                         | *                 | ×                         | ×                         | Α           | ×                         | *                     | Α           |  |

[Legend]

A: Acceptable

<sup>\*:</sup> Only programming is acceptable for the areas other than the erasure-suspended block

<sup>×:</sup> Not acceptable

#### 25.6.3 FCU Command Usage

This section shows examples of user processing procedures for firmware transfer to the FCU RAM and the issuing of FCU commands. In some procedures given in this section, the FCU state is not checked before an FCU command is issued but the command result is checked before the processing is completed. To make sure that the FCU accepts a command, check the FCU state before starting processing (see section 25.6.2, Conditions for FCU Command Acceptance).

In a flow used in this section, the current state of FCU command handling and error occurrence is checked via the FRDY, ILGLERR, ERSERR, PRGERR, SUSRDY, ERSSPD, and PRGSPD bits in FSTATR0 and the FCUERR, FRDTCT, and FRCRCT bits in FSTATR1. Since both FSTATR0 and FSTATR1 can be read in word access at a time, the FCU state can be checked by making register access only once. If the FCU state is checked via the FRDY bit of FSTATR0 and the CMDLK bit of FASTAT, register access must be made twice. However, the state of error occurrence can be checked via the CMDLK bit only.

The FRDY bit retains 0, if the FRDTCT and FRCRCT bits are set to 1 to put the FCU into a command-locked state in the middle of its command handling while the FCUERR bit is 1 or the FRDCLE and FRCCLE bits are 1. Since the FCU in a command-locked state halts its processes, the FRDY bit is never set to 1 from 0. If the FRDY retains 0 for a longer period than programming/erasing time or suspend delay time (see section 32, Electrical Characteristics), abnormal operation such as the FCU process halt may have occurred. In such case, initialize the FCU by an FCU reset. If the FRDY is set to 1 upon completion of the FCU command handling, while the FRDCLE and FRCCLE bits are 1, the FCUERR bit and the FRDTCT and FRCRCT bits are also 0. Therefore, the state of error occurrence can be checked via the ILGLERR, ERSERR, and PRGERR bits.

### (1) Transferring Firmware to the FCU RAM

To use FCU commands, the FCU firmware must be stored in the FCU RAM. When this LSI is started, the FCU firmware is not stored in the FCU RAM; copy the firmware stored in the FCU firmware area to the FCU RAM. If the FCUERR, FRDTCT, or FRCRCT bit in FSTATR1 is 1, the firmware stored in the FCU RAM may have been damaged; reset the FCU and copy the FCU firmware again in this case.

Figure 25.14 shows the procedure for firmware transfer to the FCU RAM. Before writing data to the FCU RAM, clear FENTRYR to H'0000 to stop the FCU. For details on the DMAC settings, refer to section 11, Direct Memory Access Controller (DMAC).

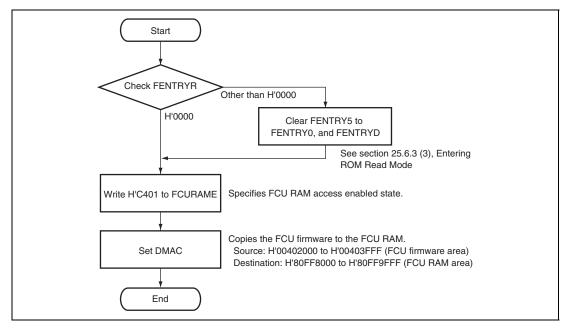


Figure 25.14 Procedure for Firmware Transfer to FCU RAM

#### (2) Entering ROM P/E Mode

To execute ROM-related FCU commands, set the FENTRY5 to FENTRY0 bits in FENTRYR appropriately to make the FCU enter ROM P/E mode (see section 25.6.2, Conditions for FCU Command Acceptance). To execute FCU commands for the ROM, set the corresponding FENTRY5 to FENTRY0 bits to 1. For the conditions for writing to the FENTRY5 to FENTRY0 bits, refer to section 25.3.10, Flash P/E Mode Entry Register (FENTRYR).

After a transition from ROM read mode to ROM P/E mode, the FCU is in ROM P/E normal mode.

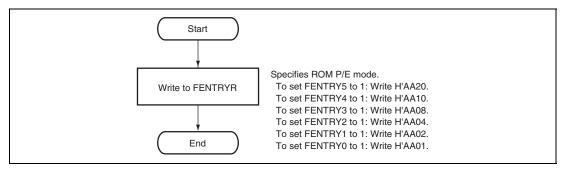


Figure 25.15 Procedure for Transition to ROM P/E Mode

### (3) Entering ROM Read Mode

To enable high-speed ROM read access through the ROM cache, clear the FENTRY5 to FENTRY0 bits in FENTRYR to make the FCU enter ROM read mode (see section 25.6.2, Conditions for FCU Command Acceptance). A transition from ROM P/E mode to ROM read mode must be made while no FCU error has been detected since FCU command processing is completed.

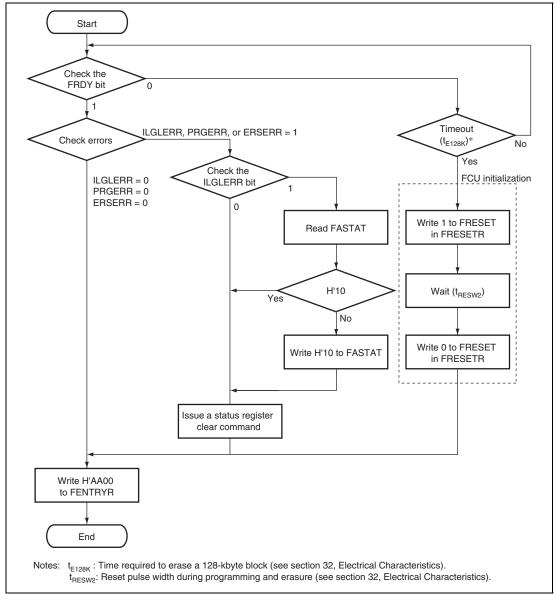


Figure 25.16 Procedure for Transition to ROM Read Mode

# (4) Using ROM P/E Normal Mode Transition Command

The FCU can be moved to ROM P/E normal mode in two ways: one is to set FENTRYR appropriately in ROM read mode (see section 25.6.3 (1), Transferring Firmware to the FCU RAM) and the other is to issue a normal mode transition command in ROM P/E mode (figure 25.17). The status read mode transition command and the lock bit read mode transition command can be used in the same way as the normal mode transition command.

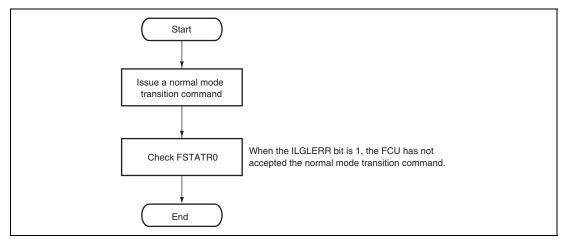


Figure 25.17 Procedure to Use ROM P/E Normal Mode Transition Command

## (5) Programming

To program the ROM, use the program command. Write byte H'E8 to a ROM program/erase address in the first cycle of the program command and byte H'80 in the second cycle. Access the P bus in words from the third to 130th cycles of the command. In the third cycle, write the programming data to the start address of the target programming area. Here, the start address must be a 256-byte boundary address. After writing words to ROM program/erase addresses 127 times, write byte H'D0 to a ROM program/erase address in the 131st cycle; the FCU then starts ROM programming. Read the FRDY bit in FSTATR0 to confirm that ROM programming is completed.

The addresses that can be specified in the first to 131st cycles depend on the setting of FENTRY5 to FENTRY0 bits in FENTRYR. An address in the range from H'80B80000 to H'80BBFFFF can be specified when the FENTRY5 bit is set to 1, an address in the range from H'80B40000 to H'80B7FFFF can be specified when the FENTRY4 bit is set to 1, an address in the range from H'80B00000 to H'80B3FFFF can be specified when the FENTRY3 bit is set to 1, an address in the range from H'80A00000 to H'80AFFFFF can be specified when the FENTRY2 bit is set to 1, an address in the range from H'80900000 to H'809FFFFF can be specified when the FENTRY1 bit is set to 1, or an address in the range from H'80800000 to H'808FFFFF can be specified when the FENTRY0 bit is set to 1. If a command is issued while an illegal combination of FENTRY5 to FENTRY0 bit values and addresses is specified, the FCU detects an error and enters command-locked state (see section 25.8.3, Error Protection).

If the area accessed in the third to 130th cycles includes addresses that do not need to be programmed, write H'FFFF as the programming data for those addresses. To ignore the protection provided by the lock bit during programming, set the FPROTCN bit in FPROTR to 1 before starting programming.

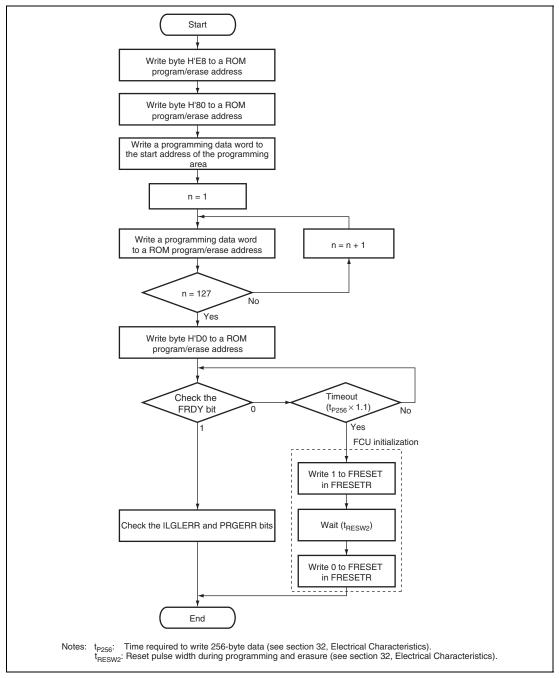


Figure 25.18 Procedure for ROM Programming

## (6) Erasure

To erase the ROM, use the block erase command. Write byte H'20 to a ROM program/erase address in the first cycle of the block erase command. Write byte H'D0 to an address in the target erasure block in the second cycle; the FCU then starts ROM erasure. Read the FRDY bit in FSTATR0 to confirm that ROM erasure is completed.

To ignore the protection provided by the lock bit during erasure, set the FPROTCN bit in FPROTR to 1 before starting erasure.

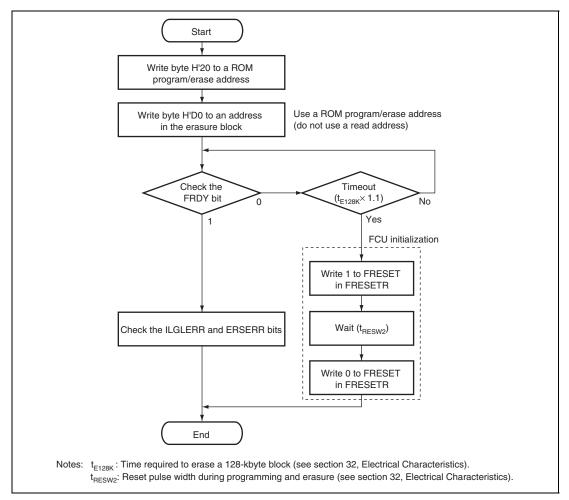


Figure 25.19 Procedure for ROM Erasure

## (7) Suspending Programming or Erasure

Page 1304 of 1824

To suspend programming or erasure of the ROM, use the P/E suspend command. Before issuing a P/E suspend command, check that the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FCUERR, FRDTCT, and FRCRCT bits in FSTATR1 are 0; that is, to ensure that programming or erasure processing is being performed correctly. Also, check that the SUSRDY bit in FSTATR1 is 1 to ensure that a suspend command is acceptable.

After issuing a P/E suspend command, read both FSTATR0 and FSTATR1 to ensure no error has occurred. If an error has occurred, at least one of the ILGLERR, PRGERR, ERSERR, FCUERR, FRDTCT, and FRCRCT bits is set to 1. If programming/erasure is complete within the period from when the SUSRDY bit is ensured to be 1 until a P/E suspend command is accepted, the ILGLERR bit is set to 1 as the issued command is detected as illegal. If a P/E suspend command is accepted when programming/erasure is complete, no error occurs, hence no transition to a suspended state (the FRDY bit is 1 and both the ERSSPD and PRGSPD bits are 0).

Once a P/E suspend command is accepted and programming/erasure is normally suspended, the FCU enters a suspended state and that the FRDY bit is 1 and the ERSSPD or PRGSPD bit is 1. After issuing a P/E suspend and ensuring that the FCU has entered a suspend state, determine which operation to perform in the succeeding process. If a P/E resume command is issued in the succeeding process while the FCU has not entered a suspended state, an illegal command error occurs and the FCU enters a command-locked state (see section 25.8.3, Error Protection).

Once the FCU has entered the erasure-suspended state, blocks not for erasing can be written to. In both programming-suspended and erasure-suspended states, the FCU can be moved to ROM read mode by clearing FENTRYR.

For the operation when the FCU accepts a P/E suspend command, see section 25.6.4, Suspending Operation.

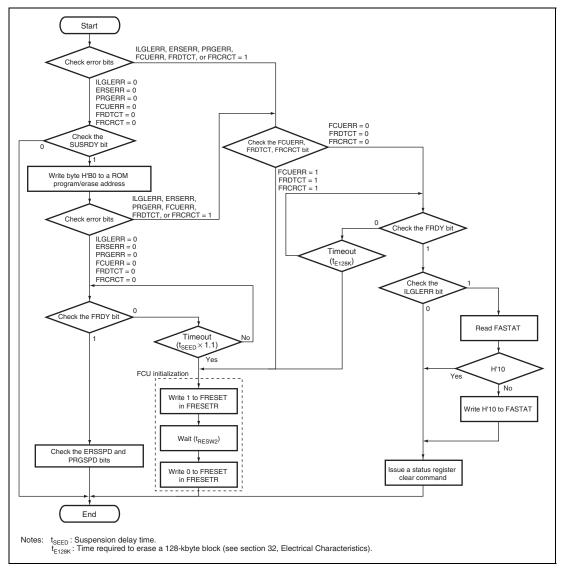


Figure 25.20 Procedure for Programming/Erasure Suspension

## (8) Resuming Programming or Erasure

To resume programming or erasure that has been suspended, use the P/E resume command. If the FENTRYR setting has been modified during suspension, issue a P/E resume command only after resetting FENTRYR to the previous value that was held before the P/E suspension command was issued.

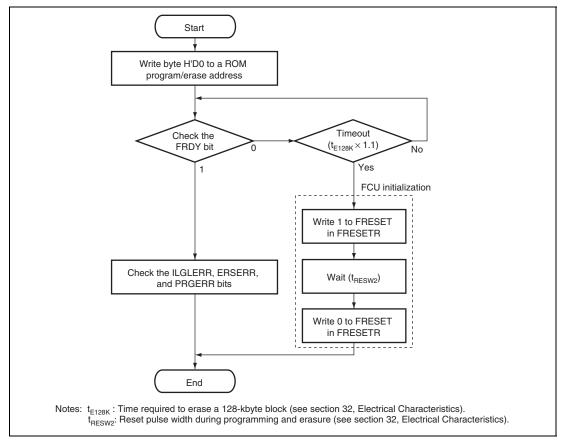


Figure 25.21 Procedure for Resuming Programming or Erasure

## (9) Clearing Status Register 0 (FSTATR0)

To clear the ILGLERR, PRGERR, and ERSERR bits in FSTATR0, use the status register clear command. When any one of the ILGLERR, PRGER, and ERSERR bits is 1, the FCU is in command-locked state, in which the FCU only accepts the status register clear command and does not accept other commands. When the ILGLERR bit is 1, check also the value of the ROMAE, EEPAE, EEPIFE, EEPRPE, and EEPWPE bits in FASTAT. If a status register clear command is issued without clearing these bits, the ILGLERR bit is not cleared.

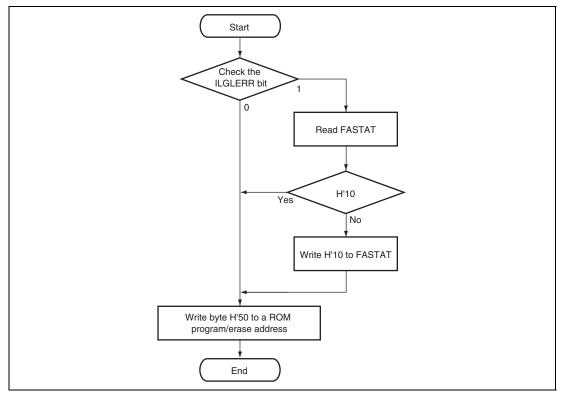


Figure 25.22 Procedure for Clearing Status Register 0

## (10) Checking Status Register 0 (FSTATR0)

The FSTATR0 value can be checked in two ways: one is to directly read FSTATR0 and the other is to read a ROM program/erase address in ROM status read mode. After an FCU command is issued that is neither a normal mode transition command nor a lock bit read mode transition command, the FCU is in ROM status read mode. In the example shown in figure 25.23, a status read mode transition command is issued to enter ROM status read mode, and then a ROM program/erase address is read to check the FSTATR0 value.

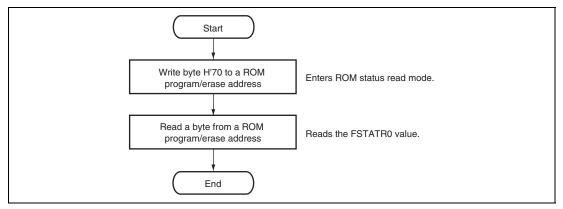


Figure 25.23 Procedure for Checking Status Register 0

## (11) Reading Lock Bit

Each erasure block in the user MAT has a lock bit. While the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be programmed or erased.

The lock bit status can be checked in either memory area read mode or register read mode. In memory area read mode (the FRDMD bit in FMODR is 0), read a ROM program/erase address in ROM lock bit read mode, and the lock bit value in the specified erasure block is copied to all bits in the data read through the P bus. In register read mode (the FRDMD bit in FMODR is 1), issue a lock bit read 2 command, and the lock bit value in the specified erasure block is copied to the FLOCKST bit in FSTATR1.

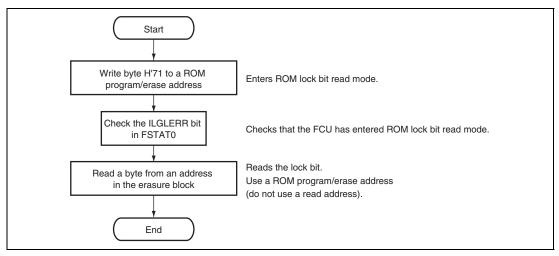


Figure 25.24 Procedure for Reading Lock Bit in Memory Area Read Mode

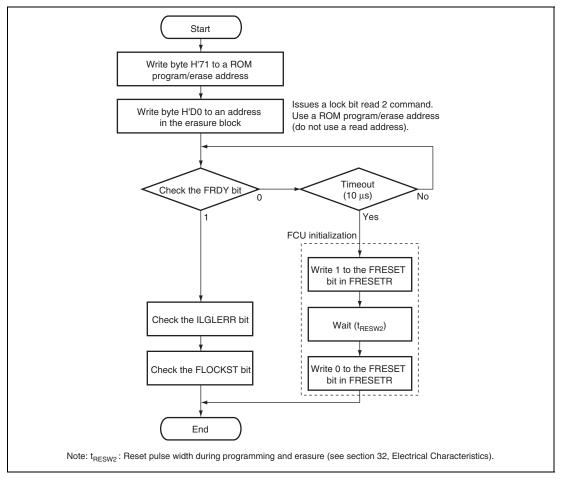


Figure 25.25 Procedure for Reading Lock Bit in Register Read Mode

## (12) Writing to Lock Bit

Each erasure block in the user MAT has a lock bit. To write to a lock bit, use the lock bit program command. Write byte H'77 to a ROM program/erase address in the first cycle of the lock bit program command. Write byte H'D0 to an address in the target erasure block whose lock bit is to be written to in the second cycle; the FCU then starts writing to the lock bit. Read the FRDY bit in FSTATR0 to confirm that writing is completed.

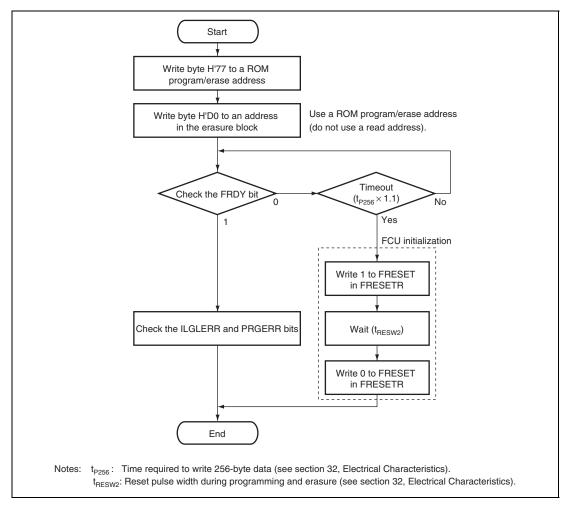


Figure 25.26 Procedure for Writing to the Lock Bit

To erase a lock bit, use the block erase command. While the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be erased. Set the FPROTCN bit to 1, and then issue a block erase command to erase a lock bit. The block erase command erases all data in the specified erasure block; it is not possible to erase only the lock bit.

## 25.6.4 Suspending Operation

When a P/E suspend command is issued while ROM is being programmed or erased, the FCU suspends the programming or erasure processing. Figure 25.27 gives an overview of operation for suspending programming. Upon accepting a programming command, the FCU clears the FRDY bit in FSTATR0 to 0 and starts programming. Once the FCU enters a state where it is ready to accept a command after the start of programming, the SUSRDY bit is set to 1. If a P/E suspend command is issued, the FCU accepts the command and clears the SUSRDY bit. If the FCU accepts the command while reapplying a write pulse, the FCU continues applying the pulse. After a specified pulse application time has elapsed, the FCU completes applying the pulse, suspends programming, and sets the PRGSPD bit to 1. Once the process completes, the FCU sets the FRDY bit to 1 and enters a programming suspended state. If the FCU accepts a P/E resume command in this state, the FCU clears the FRDY and PRGSPD bits to 0 and restarts programming.

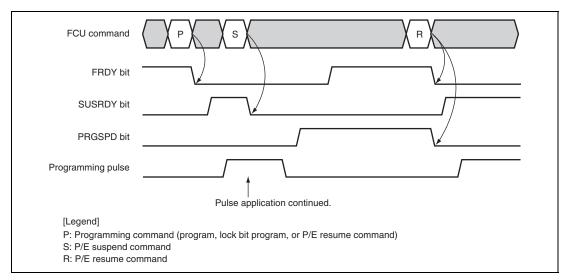


Figure 25.27 Suspending Programming Processing

Figure 25.28 shows the operation for suspending erasure processing in suspension-priority mode (the ESUSPMD bit in FCPSR is 0). Upon accepting an erasing command, the FCU clears the FRDY bit to 0 and starts erasing. Once the FCU enters a state where it is ready to accept a command after the start of erasing, the SUSRDY bit is set to 1. If a P/E suspend command is

issued, the FCU accepts the command and clears the SUSRDY bit. If the FCU accepts the command during its erasing operation, the FCU starts a suspending process even while applying a pulse and sets the ERSSPD bit to 1. Once the suspending process completes, the FCU sets the FRDY bit to 1 and enters an erasing suspended state. If the FCU accepts a P/E resume command in this state, the FCU clears the FRDY and PRGSPD bits to 0 and restarts erasing. The operations of the FRDY, SUSRDY, and ERSSPD bits are independent of the erasure-suspended mode.

The setting for the erasure-suspended mode affect the control methods for erasure pulse. In suspend-priority mode, if the FCU accepts a P/E suspend command while applying erasure pulse A, which has not been suspended previously, the FCU suspends the pulse application and enters an erasure-suspended state. After the FCU resumes erasing by accepting a P/E resume command, if the FCU accepts a P/E suspend command while applying erasing pulse A, the FCU continues applying the pulse. After a specified pulse application time has elapsed, the FCU completes applying the pulse and enters an erasure-suspended state. Next, after the FCU accepts a P/E resume command and starts applying a new pulse B, if the FCU accepts a P/E suspend command, the FCU suspends the pulse application. In suspense-priority mode, the suspense process is given priority by suspending once every pulse application.

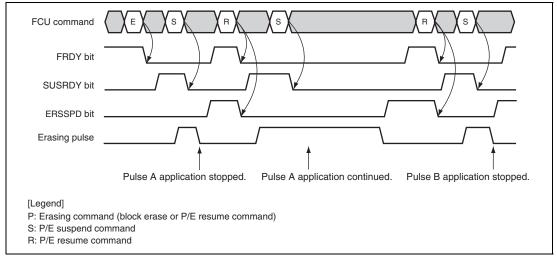


Figure 25.28 Suspending Erasure Processing (Suspension-Priority Mode)

The operation for suspending erasure processing in erasure-priority mode (the ESUSPMD bit in FCPSR is 1) is equivalent to that for suspending programming processing. In erasure-priority mode, if the FCU accepts a P/E suspend command while applying an erasing pulse, the FCU always continues applying the pulse. As processing to reapply an erasing pulse never takes place in this mode, the total time required for erasure processing is shorter than in suspension-priority mode.

Page 1313 of 1824

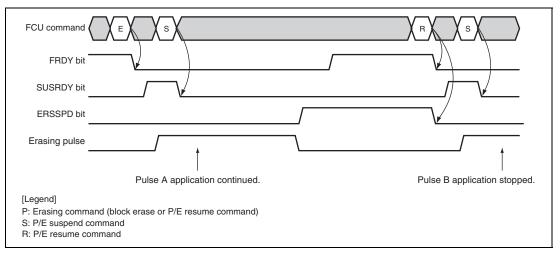


Figure 25.29 Suspending Erasure Processing (Erasure-Priority Mode)

## 25.7 User Boot Mode

To program or erase the user MAT in user boot mode, issue FCU commands to the FCU. A user-defined boot mode can be implemented by writing to the user boot MAT a ROM programming/erasing routine that uses a desired communications interface; when this LSI is started in user boot mode after that, the user-defined boot mode is initiated. Programming/erasure of the user boot MAT is only enabled in boot mode.

#### 25.7.1 User Boot Mode Initiation

When this LSI is started in user boot mode, execution starts in the embedded program stored MAT, necessary processing such as FCU firmware transfer to the FCU RAM is performed, and then execution jumps to the location indicated by the reset vector of the user boot MAT. Figure 25.30 gives an overview of the boot sequence.

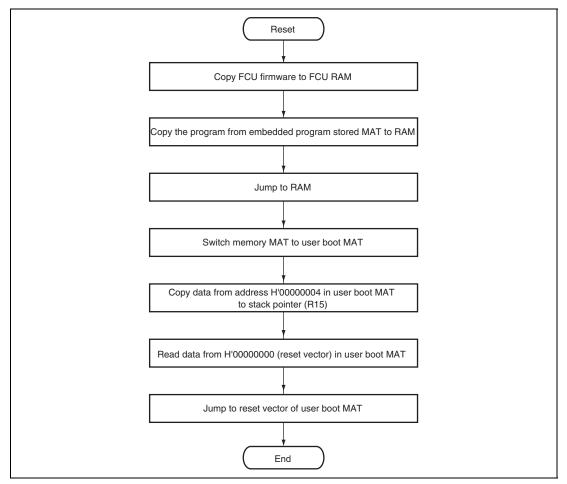


Figure 25.30 Overview of Boot Sequence in User Boot Mode

## 25.7.2 User MAT Programming

The user MAT can be programmed by starting this LSI in user boot mode while the user MAT programming/erasing routine created by the user is stored in the user boot MAT. Be sure to copy the user MAT programming/erasing routine to the RAM and execute it in the RAM. The user boot MAT is selected in the initial state in user boot mode; be sure to switch the memory MAT to the user MAT before starting programming. If an FCU command for ROM programming or erasure is issued while the user boot MAT is selected, the FCU does not program or erase the ROM. Figure 25.31 shows an example of the user MAT programming procedure.

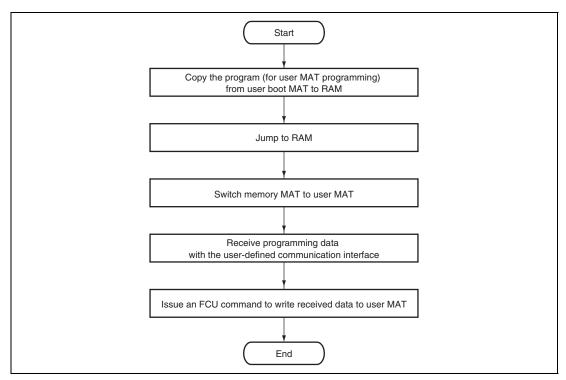


Figure 25.31 Example of User MAT Programming

## 25.8 Protection

There are three types of ROM programming/erasure protection: hardware, software, and error protection.

#### 25.8.1 Hardware Protection

The hardware protection function disables ROM programming and erasure according to the LSI pin settings.

## (1) Protection through FWE Pin

When a low level is applied to the FWE pin, the FWE bit in FPMON becomes 0. In this state, 11 cannot be written to the FENTRY5 to FENTRY0 bits in FENTRYR; that is, ROM P/E mode cannot be entered, which prevents the ROM from being programmed or erased.

When the FRDY bit is 1 and the FWE pin is driven low, the FCU clears the FENTRY5 to FENTRY0 bits to disable ROM programming and erasure. If the FRDY bit in FSTATR0 has already been set to 0 before the FWE pin is driven low, the FCU continues command processing. Even while processing a command, the FCU can accept a P/E suspend command. To resume programming or erasing the ROM, reset the FENTRY5 to FENTRY0 bits to the value that was set before being cleared, and then issue a P/E resume command.

If an attempt is made to issue a programming or erasing command to the ROM against the protection through the FWE pin, the FCU detects an error and enters command-locked state.

# (2) Protection through Mode Pins

While the on-chip ROM is disabled, ROM programming, erasing, and reading are disabled. For the operating modes set through the mode pins of this LSI, refer to section 3, Operating Modes. In user boot mode or user program mode, the user boot MAT cannot be programmed or erased.

#### 25.8.2 Software Protection

The software protection function disables ROM programming and erasure according to the control register settings or the lock bit settings in the user MAT. If an attempt is made to issue a programming or erasing command to the ROM against software protection, the FCU detects an error and enters command-locked state.

## (1) Protection through FENTRYR

When the FENTRY5 bit in FENTRYR is 0, the 0.25-Mbyte ROM (read addresses: H'00380000 to H'003BFFFF; program/erase addresses: H'80B80000 to H'80BBFFFF) is set to ROM read mode. When the FENTRY4 is 0, the 0.25-Mbyte ROM (read addresses: H'00340000 to H'0037FFFF; program/erase addresses: H'80B40000 to H'80B7FFFF) is set to ROM read mode.

When the FENTRY3 bit is 0, the 0.25-Mbyte ROM (read addresses: H'00300000 to H'0033FFFF; program/erase addresses: H'80B00000 to H'80B3FFFF) is set to ROM read mode.

When the FENTRY2 bit is 0, the 1-Mbyte ROM (read addresses: H'00200000 to H'002FFFFF; program/erase addresses: H'80A00000 to H'80AFFFF) is set to ROM read mode. When the FENTRY1 bit is 0, the 1-Mbyte ROM (read addresses: H'00100000 to H'001FFFFF; program/erase addresses: H'80900000 to H'809FFFFF) is set to ROM read mode. When the FENTRY0 bit is 0, the 1-Mbyte ROM (read addresses: H'000000000 to H'000FFFFF; program/erase addresses: H'80800000 to H'808FFFFF) is set to ROM read mode. In ROM read mode, the FCU does not accept commands, so ROM programming and erasure are disabled. If an attempt is made to issue an FCU command in ROM read mode, the FCU detects an illegal command error and enters command-locked state (see section 25.8.3, Error Protection).

## (2) Protection through Lock Bits

Each erasure block in the user MAT has a lock bit. When the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be programmed or erased. To program or erase the erasure block whose lock bit is 0, set the FPROTCN bit to 1. If an attempt is made to issue a programming or erasing command against protection by lock bits, the FCU detects an programming/erasure error and enters command-locked state (see section 25.8.3, Error Protection).

#### 25.8.3 Error Protection

The error protection function detects an illegal FCU command issued, an illegal access, or an FCU malfunction, and disables FCU command acceptance (command-locked state). While the FCU is in command-locked state, the ROM cannot be programmed or erased. To cancel command-locked state, issue a status register clear command while FASTAT is H'10.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFE) interrupt is generated if the FCU enters command-locked state (the CMDLK bit in FASTAT becomes 1). While the ROMAEINT bit in FAEINT is 1, an FIFE interrupt is generated if the ROMAE bit in FASTAT becomes 1.

Table 25.13 shows the error protection types dedicated for the ROM, those used in common by the ROM and the EEPROM, and the status bit values (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0, the FCUERR, FRDTCT, and FRCRCT bits in FSTATR1, and the ROMAE bit in FASTAT) after each error detection. If the FCU enters command-locked state due to a command other than a suspend command issued during programming or erasure processing, the FCU continues programming or erasing the ROM. In this state, the P/E suspend command cannot suspend programming or erasure. If a command is issued in command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to the previous error detection.

**Table 25.13 Error Protection Types** 

Page 1320 of 1824

| Error                 | Description  | ILGLERR | ERSERR | PRGERR | FCUERR | FRDTCT | FRCRCT | ROMAE |
|-----------------------|--|---------|--------|--------|--------|--------|--------|-------|
| FENTRYR setting error | The value set in FENTRYR is not H'0001, H'0002, H'0004, H'0008, H'0010, H'0020, or H'0080. | 1       | 0      | 0      | 0      | 0      | 0      | 0     |
|                       | The FENTRYR setting for resuming operation does not match that for suspending operation.   | 1       | 0      | 0      | 0      | 0      | 0      | 0     |

| _                 |  | GLERR. | ERSERR | PRGERR | FCUERR | FRDTCT | RCRCT | ROMAE |
|-------------------|--|--------|--------|--------|--------|--------|-------|-------|
| Illegal           | An undefined code has been specified in the first  | 1      | 0      | 0      | 0      | 0      | 0     | 0     |
| command<br>error  | cycle of an FCU command.  The value specified in the last of the multiple cycles of an FCU command is not H'D0.  | 1      | 0      | 0      | 0      | 0      | 0     | 0     |
|                   | The command issued during programming or erasure is not a suspend command.   | 1      | 0      | 0      | 0      | 0      | 0     | 0     |
|                   | A suspend command has been issued during operation that is neither programming nor erasure.  | 1      | 0      | 0      | 0      | 0      | 0     | 0     |
|                   | A suspend command has been issued in suspended state.  | 1      | 0      | 0      | 0      | 0      | 0     | 0     |
|                   | 1  | 0      | 0      | 0      | 0      | 0      | 0     |       |
|                   | 1  | 0      | 0      | 0      | 0      | 0      | 0     |       |
|                   | A block erase command has been issued in erasure-<br>suspended state.  | 1      | 0      | 0      | 0      | 0      | 0     | 0     |
|                   | 1  | 0      | 0      | 0      | 0      | 0      | 0     |       |
|                   | The value specified in the second cycle of a program command is not H'80.  | 1      | 0      | 0      | 0      | 0      | 0     | 0     |
|                   | A command has been issued in command-locked state.   | 1      | 0/1    | 0/1    | 0/1    | 0/1    | 0/1   | 0/1   |
| Erasure error     | An error has occurred during erasure processing.   | 0      | 1      | 0      | 0      | 0      | 0     | 0     |
|                   | A block erase command has been issued for the erasure block whose lock bit is set to 0 while the FPROTCN bit in FPROTR is 0.                           | 0      | 1      | 0      | 0      | 0      | 0     | 0     |
| Programming error | An error has occurred during programming processing.   | 0      | 0      | 1      | 0      | 0      | 0     | 0     |
|                   | A program, lock bit program, or program command has been issued for the erasure block whose lock bit is set to 0 while the FPROTCN bit in FPROTR is 0. | 0      | 0      | 1      | 0      | 0      | 0     | 0     |

| Error                | Description   | ILGLERR | ERSERR | PRGERR | FCUERR | FRDTCT | FRCRCT | ROMAE |
|----------------------|---|---------|--------|--------|--------|--------|--------|-------|
| FCU error            | An error has occurred during CPU processing in the FCU.   | 0       | 0      | 0      | 1      | 0      | 0      | 0     |
| FCU RAM<br>ECC error | An ECC 1-bit error has been corrected during FCU RAM reading.   | 0       | 0      | 0      | 0      | 0      | 1      | 0     |
|                      | An ECC 2-bit has been detected during FCU RAM reading.  | 0       | 0      | 0      | 1      | 1      | 0      | 0     |
| ROM access<br>error  | A read access command has been issued to addresses H'80B80000 to H'80BBFFFF while FENTRY5 = 1 in ROM P/E normal mode. | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | A read access command has been issued to addresses H'80B40000 to H'80B7FFFF while FENTRY4 = 1 in ROM P/E normal mode. | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | A read access command has been issued to addresses H'80B00000 to H'80B3FFFF while FENTRY3 = 1 in ROM P/E normal mode. | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | A read access command has been issued to addresses H'80A00000 to H'80AFFFFF while FENTRY2 = 1 in ROM P/E normal mode. | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | A read access command has been issued to addresses H'80900000 to H'809FFFF while FENTRY1 = 1 in ROM P/E normal mode.  | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | A read access command has been issued to addresses H'80800000 to H'808FFFFF while FENTRY0 = 1 in ROM P/E normal mode. | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | An access command has been issued to addresses H'80B80000 to H'80BBFFFF while FENTRY5 = 0                             | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | An access command has been issued to addresses H'80B40000 to H'80B7FFFF while FENTRY4 = 0                             | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | An access command has been issued to addresses H'80B00000 to H'80B3FFFF while FENTRY3 = 0                             | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | An access command has been issued to addresses H'80A00000 to H'80AFFFFF while FENTRY2 = 0                             | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                      | An access command has been issued to addresses H'80900000 to H'809FFFFF while FENTRY1 = 0                             | 1       | 0      | 0      | 0      | 0      | 0      | 1     |

| Error            | Description  | ILGLERR | ERSERR | PRGERR | FCUERR | FRDTCT | FRCRCT | ROMAE |
|------------------|--|---------|--------|--------|--------|--------|--------|-------|
| ROM access error | An access command has been issued to addresses H'80800000 to H'808FFFFF while FENTRY0 = 0  | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                  | A read access command has been issued to addresses H'00000000 to H'003BFFFF while the FENTRYR register value is not H'0000.  | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                  | A ROM programming or erasing command (program, lock bit program, or block erase command) has been issued while the user boot MAT is selected.                      | 1       | 0      | 0      | 0      | 0      | 0      | 1     |
|                  | An access command has been issued to an address other than the addresses for ROM programming/erasure H'80800000 to H'80807FFF while the user boot MAT is selected. | 1       | 0      | 0      | 0      | 0      | 0      | 1     |

Notes: 1. When the FRDCLE bit in FRAMECCR is set to 1.

2. When the FRCCLE bit in FRAMECCR is set to 1.

# 25.9 Usage Notes

# 25.9.1 Switching between User MAT and User Boot MAT

The user MAT and user boot MAT are allocated to the same address area. If the ROM area is accessed during switching between the user MAT and user boot MAT, an unexpected MAT may be accessed because the number of cycles required to access the ROM area depends on the internal bus status. When the ROM cache function is enabled, the previously stored data is left in the ROM cache even after MAT switching; note that a cache hit may occur when a newly selected MAT is accessed at the same address as the data stored in the cache. To avoid such unexpected behavior, take the following steps before and after MAT switching.

# 1. Modifying interrupt settings before MAT switching

There are two ways to avoid ROM area access due to an interrupt during MAT switching: one is to specify the interrupt vector fetch destination outside the ROM area through the vector base register (VBR) setting in the CPU, and the other is to mask interrupts. Note that NMI interrupts cannot be masked in this LSI; when masking interrupts to avoid ROM area access in this LSI, design the system so that no NMI is generated during MAT switching.

Switching between MATs through a program outside the ROM area
 To avoid CPU instruction fetch in the ROM area during MAT switching, execute the MAT switching processing outside the ROM area.

- 3. Performing dummy read of ROMMAT
  - After writing to ROMMAT to switch between MATs, perform a dummy read of ROMMAT to ensure that the register write is completed.
- Flushing the ROM cache during MAT switching
   Flush all lines of the ROM cache by writing 1 to the RCF bit in RCCR (see section 27, ROM Cache (ROMC)).
- Executing five or more NOP instructions
   After the dummy read of ROMMAT, execute five or more NOP instructions.

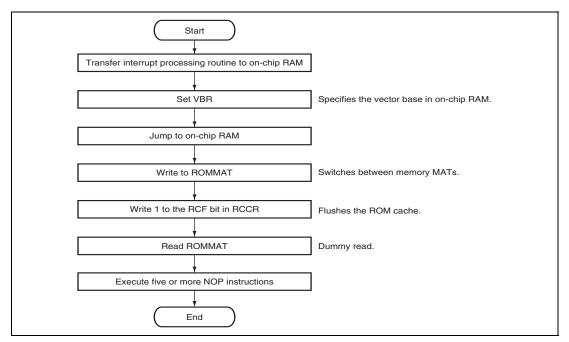


Figure 25.32 Example of MAT Switching Steps

#### 25.9.2 Other Notes

## (1) State in which AUD Operation Is Disabled and Interrupts Are Ignored

In the following modes or period, the AUD is in module standby mode and cannot operate. The NMI or maskable interrupt requests are ignored.

- Boot mode
- The program in the embedded program stored MAT is being executed immediately after the LSI is started in user boot mode

## (2) Key Code Stored Area

Addresses H'00000060 to H'00000067 in the user MAT store the key code for boot mode approval and debugging function authentication to be used with the on-chip debugger. To restrict the boot mode functions and the debugging functions, write a key code in this area. After a key code is specified through the debugger, the code is stored in this area, which should be noted during checksum verification.

## (3) Programming-/Erasure-Suspended Area

The data stored in the programming-suspended or erasure-suspended area is undetermined. To avoid malfunction due to undefined read data, ensure that no instruction is executed or no data is read from the programming-suspended or erasure-suspended area.

To avoid instruction fetch from the programming-suspended or erasure-suspended area, which may be caused by prefetch by the ROM cache, ensure that no instruction is fetched within 32 bytes from the start address of the programming-suspended or erasure-suspended area.

During ROM cache prefetch, the destination of a branch instruction is also accessed. If the destination can be in the programming-suspended or erasure-suspended area, disable the prefetch function of the ROM cache.

# (4) Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcomputers

The flash memory programming/erasing program used for conventional F-ZTAT SH microcontrollers does not work with this LSI.

## (5) FWE Pin State

Ensure that the FWE pin level does not change during programming or erasure. Even if the FWE level goes low, the current programming or erasure continues, but FENTRYR is cleared after the programming or erasure processing is completed. In this state, if an FCU command is issued without resetting FENTRYR, the FCU detects a ROM access error and enters command-locked state.

## (6) Reset during Programming or Erasure

To reset the FCU by setting the FRESET bit in the FRESETR register during programming or erasure, hold the FCU in the reset state for a period of t<sub>RESW2</sub> (see section 32, Electrical Characteristics). Since a high voltage is applied to the ROM during programming and erasure, the FCU has to be held in the reset state long enough to ensure that the voltage applied to the memory unit has dropped. Do not read from the ROM while the FCU is in the reset state.

When a power-on reset is generated by asserting the  $\overline{RES}$  pin during programming or erasure of the flash memory, hold the reset state for a period of  $t_{RESW2}$  (see section 32, Electrical Characteristics). In a power-on reset, not only does the voltage applied to the memory unit have to drop, but the power supply for the ROM and its internal circuitry also have to be initialized. Thus, the reset state must be maintained over a longer period than in the case of resetting the FCU.

While programming or erasure is performed, do not generate an internal reset caused by WDT counter overflow. A reset caused by WDT cannot ensure a sufficient time required for voltage drop for the memory unit, initialization of the power supply for the ROM, or initialization of its internal circuit.

When either a power-on reset by asserting the  $\overline{RES}$  pin, or an FCU reset by setting the FRESET bit in the FRESETR register, is executed during programming or erasure, the whole data in the programming or erasure area becomes undefined.

## (7) Prohibition of Additional Programming

One area cannot be programmed twice in succession. To program an area that has already been programmed, be sure to erase the area before reprogramming.

## (8) Suspension by Programming or Erasure Suspension

When programming or erasure is suspended by a programming or an erasure suspend command, the programming or erasure must be completed by a resume command.

## (9) Power off during Programming or Erasure

Do not switch the power off during programming or erasure. As a high voltage is applied to the ROM during programming and erasure, a certain period is required for this voltage to fall.

Thus, to allow for cases where switching the power off during programming or erasure cannot be avoided, design the system so that at least the  $V_{cc}$  holding time at  $PV_{cc}$  shutdown ( $t_{vccH}$ ) is secured by assertion of the  $\overline{HSTBY}$  signal causing a transition to the hardware standby state if the power is switched off. For details, see section 32, Electrical Characteristics.

## (10) Prohibition of Clearing FRDCLE Bit to 0

Whenever the FRDTCT bit in FSTATR1 is set to 1, the FCU has to enter command-locked state, because the FCU command operation cannot be guaranteed. Thus, do not clear the FRDCLE bit in FRAMECCR to 0.

## (11) Note on Transition to ROM Read Mode

When transiting to ROM read mode by changing FENTRY5 to FENTRY0 bits in FENTRYR register from 1 to 0, write 0 to FENTRY5 to FENTRY0 bits and execute a minimum of 5 NOP instructions after performing a dummy read of FENTRYR register.

SH72546R Section 26 EEPROM

# Section 26 EEPROM

This LSI includes 128 Kbytes of flash memory (EEPROM) for storing data. The flash memory has the following features.

### 26.1 Features

Flash-memory MATs

The EEPROM has two types of memory areas (hereafter referred to as memory MATs) in the same address space. These two MATs can be switched by bank switching through the control register. For addresses H'80100080 to H'8011FFFF, the data MAT contents will always be read even when the product information MAT is selected. The product information MAT cannot be programmed or erased.

Data MAT: 128 Kbytes

Product information MAT: 128 bytes

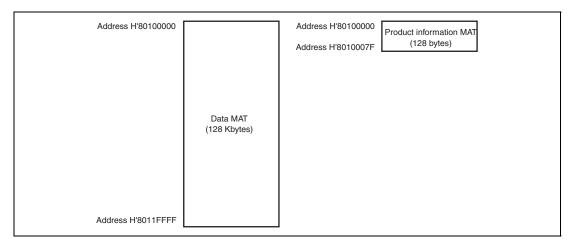


Figure 26.1 Memory MAT Configuration in EEPROM

• Reading through the peripheral bus (P bus)

Both the data MAT and product information MAT can be read through the P bus in three peripheral clock (P $\phi$ ) cycles in words or bytes, or in five P $\phi$  cycles in longwords, when P $\phi$  is 16 to 20 MHz. When P $\phi$  is 32 to 40 MHz, they can be read through the P bus in four P $\phi$  cycles in words or bytes, or in seven P $\phi$  cycles in longwords.

Section 26 EEPROM SH72546R

Programming and erasing methods

The data MAT can be programmed and erased by commands issued through the peripheral bus (P bus) to the ROM/EEPROM-dedicated sequencer (FCU).

While the FCU is programming or erasing the data MAT, the CPU can execute a program located in the ROM, RAM, or external address space. While the FCU is programming or erasing the ROM or data MAT, data cannot be read form the data MAT. When the FCU suspends programming or erasure, the CPU can read data from the data MAT, and then the FCU can resume programming or erasure of the data MAT. While the FCU suspends erasure, areas other than the erasure-suspended area can be programmed.

SH72546R Section 26 EEPROM

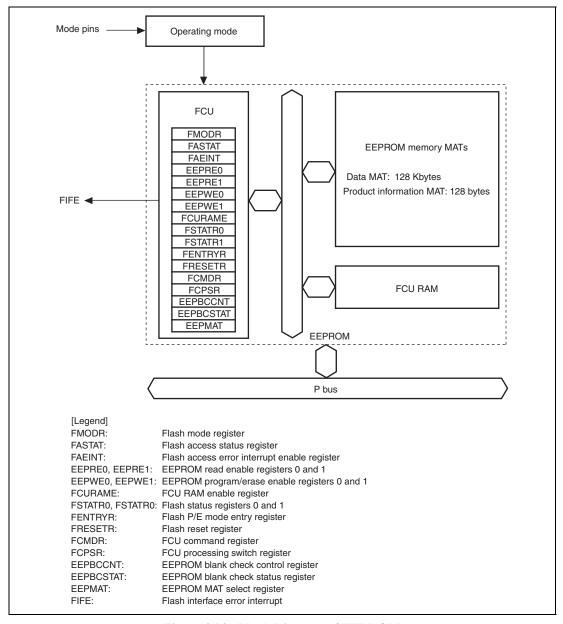


Figure 26.2 Block Diagram of EEPROM

Section 26 EEPROM SH72546R

## • Programming/erasing unit

The data MAT is programmed in 8-byte or 128-byte units and erased in block units (8 Kbytes) in user mode, user program mode, and user boot mode. In boot mode, the data MAT is programmed in 256-Kbyte units and erased in block units (8 Kbytes). The product information MAT is read-only memory and cannot be programmed or erased.

Figure 26.3 shows the block configuration of the data MAT of this LSI. The data MAT is divided into sixteen 8-Kbyte blocks (DB00 to DB15).

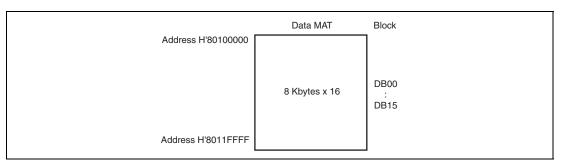


Figure 26.3 Block Configuration of Data MAT

#### Blank check function

If data is read from erased EEPROM by the CPU, undefined values are read. Using blank check command of the FCU allows checking of whether the EEPROM is erased (in a blank state). Either an 8 Kbytes (1 erasure block) or 8 bytes of area can be checked by a single execution of the blank check command.

Blank check function checks the erase state of the area where erase has ended. The function is disabled when programming/erasing was suspended (e.g., reset input, power-supply interruption).

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

The data MAT can be programmed using the SCI. The bit rate for SCI communications between the host and the LSI can be automatically adjusted.

— User mode/user program mode

The data MAT can be programmed with a desired interface. The user mode includes the MCU extended mode and MCU single-chip mode (modes 2 and 3) in which the on-chip ROM is enabled.

User boot mode

The data MAT can be programmed with a desired interface. To make a transition to this mode, a reset is needed.

SH72546R Section 26 EEPROM

## Protection modes

This LSI supports two modes to protect memory against programming, erasing, or reading: hardware protection by the levels on the mode pins and software protection by the setting of the FENTRYD bit, EEPRE0 and EEPRE1 registers, or EEPWE0 and EEPWE1 registers. The FENTRYD bit enables or disables data MAT programming or erasure by the FCU. EEPRE0 and EEPRE1 control protection of each data MAT block against reading, and EEPWE0 and EEPWE1 control protection against programming and erasure.

The LSI also provides a function to suspend programming or erasure when abnormal operation is detected during programming or erasure. In addition, the LSI provides a function to protect the EEPROM against instruction fetch attempted by the CPU.

Programming and erasing time and count
 Refer to section 32, Electrical Characteristics.

Section 26 EEPROM SH72546R

# 26.2 Input/Output Pins

Table 26.1 shows the input/output pins used for the EEPROM. The combination of MD4 to MD0 pin levels determines the EEPROM programming mode (see section 26.4, Overview of EEPROM-Related Modes). In boot mode, programming and erasing the EEPROM can be performed by the host via the pins PJ6/RxD\_A and PJ5/TxD\_A (refer to section 26.5, Boot Mode).

**Table 26.1 Pin Configuration** 

| Pin Name                       | Symbol     | I/O    | Function  |
|--------------------------------|------------|--------|---|
| Power-on reset                 | RES        | Input  | This LSI enters the power-on reset state when this signal goes low. |
| Mode                           | MD4 to MD0 | Input  | These pins specify the operating mode.                              |
| Receive data in SCI channel A  | PJ6/RxD_A  | Input  | Receives data through SCI channel A (communications with host)      |
| Transmit data in SCI channel A | PJ5/TxD_A  | Output | Transmits data through SCI channel A (communications with host)     |

# **26.3** Register Descriptions

Table 26.2 shows the EEPROM-related registers. Some of these registers have ROM-related bits, but this section only describes the EEPROM-related bits. For the registers consisting of bits used by the ROM and EEPROM in common (FCURAME, FSTATR0, FSTATR1, FRESETR, FCMDR, and FCPSR) and the ROM-dedicated bits, refer to section 25.3, Register Descriptions. The EEPROM-related registers are initialized by a power-on reset or a transition to the hardware standby mode.

SH72546R Section 26 EEPROM

**Table 26.2 Register Configuration** 

| Register Name                                | Symbol    | <b>R/W</b> * <sup>1</sup>          | Initial<br>Value | Address    | Access<br>Size |
|--|-----------|------------------------------------|------------------|------------|----------------|
| Flash mode register                          | FMODR     | R/W                                | H'00             | H'FFFFA802 | 8              |
| Flash access status register                 | FASTAT    | R/(W)*2                            | H'00             | H'FFFFA810 | 8              |
| Flash access error interrupt enable register | FAEINT    | R/W                                | H'9F             | H'FFFFA811 | 8              |
| EEPROM read enable register 0                | EEPRE0    | R/(W)*3                            | H'0000           | H'FFFFA840 | 8, 16          |
| EEPROM read enable register 1                | EEPRE1    | R/(W)*3                            | H'0000           | H'FFFFA842 | 8, 16          |
| EEPROM program/erase enable register 0       | EEPWE0    | R/(W)*3                            | H'0000           | H'FFFFA850 | 8, 16          |
| EEPROM program/erase enable register 1       | EEPWE1    | R/(W)*3                            | H'0000           | H'FFFFA852 | 8, 16          |
| FCU RAM enable register                      | FCURAME   | R/(W)*3                            | H'0000           | H'FFFFA854 | 8, 16          |
| Flash status register 0                      | FSTATR0   | R*5                                | H'80             | H'FFFFA900 | 8, 16          |
| Flash status register 1                      | FSTATR1   | R*⁵                                | H'00             | H'FFFFA901 | 8, 16          |
| Flash P/E mode entry register                | FENTRYR   | R/(W)* <sup>4</sup> * <sup>5</sup> | H'0000           | H'FFFFA902 | 8, 16          |
| Flash reset register                         | FRESETR   | R/(W)*3                            | H'0000           | H'FFFFA906 | 8, 16          |
| FCU command register                         | FCMDR     | R*⁵                                | H'FFFF           | H'FFFFA90A | 8, 16          |
| FCU processing switch register               | FCPSR     | R/W* <sup>5</sup>                  | H'0000           | H'FFFFA918 | 8, 16          |
| EEPROM blank check control register          | EEPBCCNT  | R/W* <sup>5</sup>                  | H'0000           | H'FFFFA91A | 8, 16          |
| EEPROM blank check status register           | EEPBCSTAT | R* <sup>5</sup>                    | H'0000           | H'FFFFA91E | 8, 16          |
| EEPROM MAT select register                   | EEPMAT    | R/(W)*3                            | H'0000           | H'FFFFAB00 | 8, 16          |

Notes: 1. In on-chip ROM disabled mode, the bits of the EEPROM-related registers are always read as 0 and writing to them is ignored.

- 2. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.
- 3. This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.
- 4. This register can be written to only when a specified value is written to the upper byte in word access; the register is initialized when a value not allowed for the register is written to the upper byte. The data written to the upper byte is not stored in the register.
- 5. This register can be initialized by a power-on reset, a transition to the hardware standby mode, or by setting the FRESET bit of FRESETR to 1.

Section 26 EEPROM SH72546R

#### 26.3.1 Flash Mode Register (FMODR)

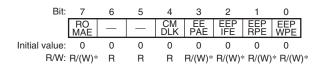
FMODR specifies an operating mode for the FCU. In on-chip ROM disabled mode, the FMODR bits are always read as H'00, and writing to them is ignored. FMODR can be initialized by a power-on reset or a transition to the hardware standby mode.

| Bit:           | 7 | 6 | 5 | 4         | 3 | 2 | 1 | 0 |
|----------------|---|---|---|-----------|---|---|---|---|
|                | _ | _ | _ | FR<br>DMD | _ | _ | _ | _ |
| Initial value: | 0 | 0 | 0 | 0         | 0 | 0 | 0 | 0 |
| R/W:           | R | R | R | R/W       | R | R | R | R |

| Bit    | Bit Name | Initial<br>Value | R/W | Description  |
|--------|----------|------------------|-----|--|
| 7 to 5 | DIL Name | All 0            | R   | Reserved   |
| 7 10 5 | _        | All U            | н   |  |
|        |          |                  |     | Write values must always be 0; otherwise normal operation cannot be guaranteed.  |
| 4      | FRDMD    | 0                | R/W | FCU Read Mode Select Bit   |
|        |          |                  |     | Selects the read mode to read the ROM or EEPROM using FCU. This bit specifies the EEPROM lock bit read mode transition or blank check processings in the EEPROM (see section 26.6.1, FCU Command List, 26.6.3, FCU Command Usage), whereas this bit must be set to specify the read method for the lock bits in the ROM (see section 25, ROM). |
|        |          |                  |     | 0: Memory area read mode   |
|        |          |                  |     | This mode is selected to enter the EEPROM lock bit read mode. Since the EEPROM has no lock bits, reading an EEPROM area results in an undefined value.   |
|        |          |                  |     | 1: Register read mode  |
|        |          |                  |     | To make the blank check command available for use, register read mode is set.  |
| 3 to 0 | _        | All 0            | R   | Reserved   |
|        |          |                  |     | Write values must always be 0; otherwise normal operation cannot be guaranteed.  |

## 26.3.2 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and EEPROM. In on-chip ROM disabled mode, FASTAT is read as H'00 and writing to it is ignored. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 26.7.3, Error Protection). To cancel command-locked state, set FASTAT to H'10, and then issue a status-clear command to the FCU. FASTAT is initialized by a power-on reset or a transition to the hardware standby mode.



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

| Bit  | Bit Name | Initial<br>Value | R/W    | Description  |
|------|----------|------------------|--------|--|
| 7    | ROMAE    | 0                | R/(W)* | ROM Access Error   |
|      |          |                  |        | Refer to section 25, ROM.  |
| 6, 5 | _        | All 0            | R      | Reserved   |
|      |          |                  |        | These bits are always read as 0. The write value should always be 0.                         |
| 4    | CMDLK    | 0                | R      | FCU Command Lock   |
|      |          |                  |        | Indicates whether the FCU is in command-locked state (see section 26.7.3, Error Protection). |
|      |          |                  |        | 0: The FCU is not in command-locked state  |
|      |          |                  |        | 1: The FCU is in command-locked state  |
|      |          |                  |        | [Setting condition]  |
|      |          |                  |        | <ul> <li>The FCU detects an error and enters command-<br/>locked state.</li> </ul>           |
|      |          |                  |        | [Clearing condition]   |
|      |          |                  |        | <ul> <li>The FCU completes the status-clear command processing.</li> </ul>                   |

| Bit | Bit Name | Initial<br>Value | R/W    | Description   |
|-----|----------|------------------|--------|---|
| 3   | EEPAE    | 0                | R/(W)* | EEPROM Access Error   |
|     |          |                  |        | Indicates whether an access error has been generated for the EEPROM. If this bit becomes 1, the ILGLERR bit in FSTATR0 is set to 1 and the FCU enters command-locked state. |
|     |          |                  |        | 0: No EEPROM access error has occurred  |
|     |          |                  |        | 1: An EEPROM access error has occurred  |
|     |          |                  |        | [Setting conditions]  |
|     |          |                  |        | <ul> <li>A read access command is issued to the<br/>EEPROM area while the FENTRYD bit in<br/>FENTRYR is 1 in EEPROM P/E normal mode.</li> </ul>                             |
|     |          |                  |        | <ul> <li>A write access command is issued to the<br/>EEPROM area while the FENTRYD bit in<br/>FENTRYR is 0.</li> </ul>  |
|     |          |                  |        | <ul> <li>An access command is issued to the EEPROM<br/>area while one of the FENTRY5 to FENTRY0 bits<br/>in FENTRYR is 1.</li> </ul>  |
|     |          |                  |        | [Clearing condition]  |
|     |          |                  |        | • 0 is written to this bit after reading EEPAE = 1.   |
| 2   | EEPIFE   | 0                | R/(W)* | EEPROM Instruction Fetch Error  |
|     |          |                  |        | Indicates whether an instruction fetch error has been generated for the EEPROM.   |
|     |          |                  |        | 0: No EEPROM instruction fetch error has occurred   |
|     |          |                  |        | 1: An EEPROM instruction fetch error has occurred   |
|     |          |                  |        | [Setting condition]   |
|     |          |                  |        | <ul> <li>An attempt is made to fetch an instruction from the<br/>EEPROM.</li> </ul>   |
|     |          |                  |        | [Clearing condition]  |
|     |          |                  |        | • 0 is written to this bit after reading EEPIFE = 1.  |

| D:1 | Dit Name | Initial | D ///  | Description  |
|-----|----------|---------|--------|--|
| Bit | Bit Name | Value   | R/W    | Description  |
| 1   | EEPRPE   | 0       | R/(W)* | EEPROM Read Protect Error  |
|     |          |         |        | Indicates whether an error has been generated against the EEPROM read protection provided by the EEPRE0 and EEPRE1 settings.   |
|     |          |         |        | 0: The EEPROM has not been read against the EEPRE0 and EEPRE1 settings   |
|     |          |         |        | An attempt has been made to read data from the EEPROM against the EEPRE0 and EEPRE1 settings   |
|     |          |         |        | [Setting condition]  |
|     |          |         |        | <ul> <li>An attempt is made to read data from the<br/>EEPROM area that has been read-protected<br/>through the EEPRE0 and EEPRE1 settings.</li> </ul>                        |
|     |          |         |        | [Clearing condition]   |
|     |          |         |        | • 0 is written to this bit after reading EEPRPE = 1.   |
| 0   | EEPWPE   | 0       | R/(W)* | EEPROM Program/Erase Protect Error   |
|     |          |         |        | Indicates whether an error has been generated against the EEPROM program/erasure protection provided by the EEPWE0 and EEPWE1 settings.                                      |
|     |          |         |        | No programming or erasing command has been issued to the EEPROM against the EEPWE0 and EEPWE1 settings   |
|     |          |         |        | <ol> <li>A programming or erasing command has been<br/>issued to the EEPROM against the EEPWE0 and<br/>EEPWE1 settings</li> </ol>  |
|     |          |         |        | [Setting condition]  |
|     |          |         |        | <ul> <li>A programming or erasing command is issued to<br/>the EEPROM area that has been program/erase-<br/>protected through the EEPWE0 and EEPWE1<br/>settings.</li> </ul> |
|     |          |         |        | [Clearing condition]   |
|     |          |         |        | • 0 is written to this bit after reading EEPWPE = 1.   |

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

#### 26.3.3 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupt requests. In on-chip ROM disabled mode, FAEINT is read as H'00 and writing to it is ignored. FAEINT is initialized by a power-on reset or a transition to the hardware standby mode.

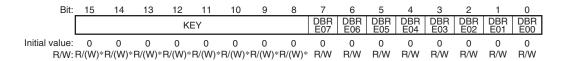
| Bit:           | 7           | 6 | 5 | 4           | 3           | 2            | 1            | 0            |
|----------------|-------------|---|---|-------------|-------------|--------------|--------------|--------------|
|                | ROM<br>AEIE | _ | _ | CMD<br>LKIE | EEP<br>AEIE | EEPI<br>FEIE | EEPR<br>PEIE | EEPW<br>PEIE |
| Initial value: | 1           | 0 | 0 | 1           | 1           | 1            | 1            | 1            |
| R/W:           | R/W         | R | R | R/W         | R/W         | R/W          | R/W          | R/W          |

| Bit  | Bit Name | Initial<br>Value | R/W | Description   |
|------|----------|------------------|-----|---|
| 7    | ROMAEIE  | 1                | R/W | ROM Access Error Interrupt Enable   |
|      |          |                  |     | Refer to section 25, ROM.   |
| 6, 5 | _        | All 0            | R   | Reserved  |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 4    | CMDLKIE  | 1                | R/W | FCU Command Lock Interrupt Enable   |
|      |          |                  |     | Enables or disables an FIFE interrupt request when FCU command-locked state is entered and the CMDLK bit in FASTAT becomes 1. |
|      |          |                  |     | 0: Does not generate an FIFE interrupt request when CMDLK = 1   |
|      |          |                  |     | 1: Generates an FIFE interrupt request when CMDLK = 1   |
| 3    | EEPAEIE  | 1                | R/W | EEPROM Access Error Interrupt Enable  |
|      |          |                  |     | Enables or disables an FIFE interrupt request when an EEPROM access error occurs and the EEPAE bit in FASTAT becomes 1.       |
|      |          |                  |     | 0: Does not generate an FIFE interrupt request when EEPAE = 1   |
|      |          |                  |     | 1: Generates an FIFE interrupt request when EEPAE = 1   |

|     |          | Initial |     |   |
|-----|----------|---------|-----|---|
| Bit | Bit Name | Value   | R/W | Description   |
| 2   | EEPIFEIE | 1       | R/W | EEPROM Instruction Fetch Error Interrupt enable   |
|     |          |         |     | Enables or disables an FIFE interrupt request when an EEPROM instruction fetch error occurs and the EEPIFE bit in FASTAT becomes 1.     |
|     |          |         |     | 0: Does not generate an FIFE interrupt request when EEPIFE = 1  |
|     |          |         |     | 1: Generates an FIFE interrupt request when EEPIFE = 1  |
| 1   | EEPRPEIE | 1       | R/W | EEPROM Read Protect Error Interrupt Enable  |
|     |          |         |     | Enables or disables an FIFE interrupt request when an EEPROM read protect error occurs and the EEPRPE bit in FASTAT becomes 1.          |
|     |          |         |     | 0: Does not generate an FIFE interrupt request when<br>EEPRPE = 1   |
|     |          |         |     | 1: Generates an FIFE interrupt request when<br>EEPRPE = 1   |
| 0   | EEPWPEIE | 1       | R/W | EEPROM Program/Erase Protect Error Interrupt<br>Enable  |
|     |          |         |     | Enables or disables an FIFE interrupt request when an EEPROM program/erase protect error occurs and the EEPWPE bit in FASTAT becomes 1. |
|     |          |         |     | 0: Does not generate an FIFE interrupt request when EEPWPE = 1  |
|     |          |         |     | 1: Generates an FIFE interrupt request when EEPWPE = 1  |

#### **EEPROM Read Enable Register 0 (EEPRE0)** 26.3.4

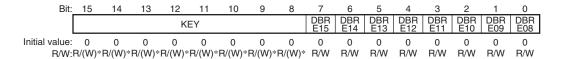
EEPRE0 enables or disables read access to blocks DB00 to DB07 (see figure 26.3) in the data MAT. In on-chip ROM disabled mode, EEPRE0 is read as H'0000 and writing to it is ignored. EEPRE0 is initialized by a power-on reset or a transition to the hardware standby mode.



| Bit     | Bit Name | Initial<br>Value | R/W    | Description  |
|---------|----------|------------------|--------|--|
| 15 to 8 | KEY      | H'00             | R/(W)* | Key Code   |
|         |          |                  |        | These bits enable or disable DBRE07 to DBRE00 bit modification. The data written to these bits are not stored. |
| 7       | DBRE07   | 0                | R/W    | DB07 to DB00 Block Read Enable   |
| 6       | DBRE06   | 0                | R/W    | Enables or disables read access to blocks DB07 to  |
| 5       | DBRE05   | 0                | R/W    | DB00 in the data MAT. The DBREi bit (i = 07 to 00)  controls read access to block DBi. Writing to these        |
| 4       | DBRE04   | 0                | R/W    | bits is enabled only when this register is accessed in   |
| 3       | DBRE03   | 0                | R/W    | word size and H'2D is written to the KEY bits.   |
| 2       | DBRE02   | 0                | R/W    | 0: Disables read access  |
| 1       | DBRE01   | 0                | R/W    | 1: Enables read access   |
| 0       | DBRE00   | 0                | R/W    | _  |

## **26.3.5** EEPROM Read Enable Register 1 (EEPRE1)

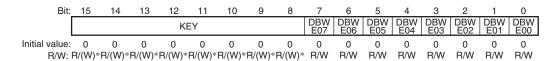
EEPRE1 enables or disables read access to blocks DB08 to DB15 (see figure 26.3) in the data MAT. In on-chip ROM disabled mode, EEPRE1 is read as H'0000 and writing to it is ignored. EEPRE1 is initialized by a power-on reset or a transition to the hardware standby mode.



| Bit     | Bit Name | Initial<br>Value | R/W    | Description  |
|---------|----------|------------------|--------|--|
| 15 to 8 | KEY      | H'00             | R/(W)* | Key Code   |
|         |          |                  |        | These bits enable or disable DBRE15 to DBRE08 bit modification. The data written to these bits are not stored. |
| 7       | DBRE15   | 0                | R/W    | DB15 to DB08 Block Read Enable   |
| 6       | DBRE14   | 0                | R/W    | Enables or disables read access to blocks DB15 to  |
| 5       | DBRE13   | 0                | R/W    | DB08 in the data MAT. The DBREi bit (i = 15 to 08)  controls read access to block DBi.                         |
| 4       | DBRE12   | 0                | R/W    | _Writing to these bits is enabled only when this register  |
| 3       | DBRE11   | 0                | R/W    | is accessed in word size and H'D2 is written to the  |
| 2       | DBRE10   | 0                | R/W    | KEY bits.  |
| 1       | DBRE09   | 0                | R/W    | 0: Disables read access  |
| 0       | DBRE08   | 0                | R/W    | 1: Enables read access   |

#### EEPROM Program/Erase Enable Register 0 (EEPWE0) 26.3.6

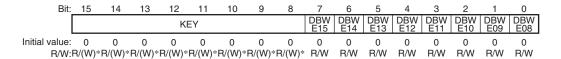
EEPWE0 enables or disables programming and erasure of blocks DB00 to DB07 (see figure 26.3) in the data MAT. In on-chip ROM disabled mode, EEPWE0 is read as H'0000 and writing to it is ignored. EEPWE0 is initialized by a power-on reset or a transition to the hardware standby mode.



| Bit     | Bit Name | Initial<br>Value | R/W    | Description  |
|---------|----------|------------------|--------|--|
| 15 to 8 | KEY      | H'00             | R/(W)* | Key Code   |
|         |          |                  |        | These bits enable or disable DBWE07 to DBWE00 bit modification. The data written to these bits are not stored. |
| 7       | DBWE07   | 0                | R/W    | DB07 to DB00 Block Program/Erase Enable  |
| 6       | DBWE06   | 0                | R/W    | Enables or disables programming and erasure of   |
| 5       | DBWE05   | 0                | R/W    | blocks DB07 to DB00 in the data MAT. The DBWEi bit (i = 07 to 00) controls programming and erasure of          |
| 4       | DBWE04   | 0                | R/W    | block DBi. Writing to these bits is enabled only when  |
| 3       | DBWE03   | 0                | R/W    | this register is accessed in word size and H'1E is written to the KEY bits.                                    |
| 2       | DBWE02   | 0                | R/W    | - Written to the KEY bits 0: Disables programming and erasure  |
| 1       | DBWE01   | 0                | R/W    | Disables programming and erasure     1: Enables programming and erasure  |
| 0       | DBWE00   | 0                | R/W    | - 1. Enables programming and erasure   |

## 26.3.7 EEPROM Program/Erase Enable Register 1 (EEPWE1)

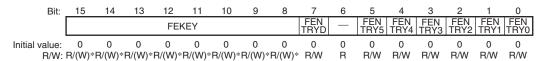
EEPWE1 enables or disables programming and erasure of blocks DB08 to DB15 (see figure 26.3) in the data MAT. In on-chip ROM disabled mode, EEPWE1 is read as H'0000 and writing to it is ignored. EEPWE1 is initialized by a power-on reset or a transition to the hardware standby mode.



| Bit     | Bit Name | Initial<br>Value | R/W    | Description  |
|---------|----------|------------------|--------|--|
| 15 to 8 | KEY      | H'00             | R/(W)* | Key Code   |
|         |          |                  |        | These bits enable or disable DBWE15 to DBWE08 bit modification. The data written to these bits are not stored. |
| 7       | DBWE15   | 0                | R/W    | DB15 to DB08 Block Program/Erase Enable  |
| 6       | DBWE14   | 0                | R/W    | Enables or disables programming and erasure of   |
| 5       | DBWE13   | 0                | R/W    | blocks DB15 to DB08 in the data MAT. The DBWEi bit if it is 15 to 08) controls read access to block DBi.       |
| 4       | DBWE12   | 0                | R/W    | _Writing to these bits is enabled only when this register  |
| 3       | DBWE11   | 0                | R/W    | is accessed in word size and H'E1 is written to the  |
| 2       | DBWE10   | 0                | R/W    | KEY bits.  |
| 1       | DBWE09   | 0                | R/W    | 0: Disables programming and erasure  |
| 0       | DBWE08   | 0                | R/W    | 1: Enables programming and erasure   |

### 26.3.8 Flash P/E Mode Entry Register (FENTRYR)

FENTRYR specifies the P/E mode for the ROM or EEPROM. To specify the P/E mode for the ROM or EEPROM so that the FCU can accept commands, set FENTRYD and FENTRY5—FENTRY0 to 1. Note that if this register is set to other than H'0001, H'0002, H'0004, H'0008, H0010, H'0020, or H'0080, the ILGLERR bit in the FSTATR0 register will be set and the FCU will enter command-locked state. In on-chip ROM disabled mode, FENTRYR is read as H'0000 and writing to it is ignored. FENTRYR is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

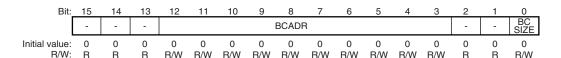


| Bit     | Bit Name | Initial<br>Value | R/W    | Description  |
|---------|----------|------------------|--------|--|
| 15 to 8 | FEKEY    | H'00             | R/(W)* | Key Code   |
|         |          |                  |        | These bits enable or disable the FENTRYD and FENTRY5 to FENTRY0 bit modification. The data written to these bits are not stored. |

| Bit    | Bit Name                 | Initial<br>Value | R/W | Description  |
|--------|--------------------------|------------------|-----|--|
| 7      | FENTRYD                  | 0                | R/W | EEPROM P/E Mode Entry This bit specifies the P/E mode for the EEPROM.  0: The EEPROM is in read mode 1: The EEPROM is in P/E mode [Write enabling conditions] When the following conditions are all satisfied:  • The LSI is in on-chip ROM enabled mode.  • The FRDY bit in FSTATR0 is 1.  • H'AA is written to FEKEY in word access. [Setting condition]  • 1 is written to FENTRYD while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions]  • This register is written to in byte access.  • A value other than H'AA is written to FEKEY in word access.  • 0 is written to FENTRYD while the write enabling conditions are satisfied.  • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied. |
| 6      | _                        | 0                | R   | Reserved  These bits are always read as 0. The write value should always be 0.   |
| 5 to 0 | FENTRY5<br>to<br>FENTRY0 | All 0            | R/W | ROM P/E Mode Entry 5 to 0<br>Refer to section 25, ROM.   |

#### **EEPROM Blank Check Control Register (EEPBCCNT)** 26.3.9

EEPBCCNT specifies the addresses and sizes of the target areas to be checked by the blank check command. In on-chip ROM disabled mode, EEPBCCNT is read as H'0000, and writing to it is ignored. EEPBCCNT is initialized by a power-on reset, a transition to the hardware standby mode, or by setting the FRESET bit of FRESETR to 1.



| D:4      | Bit Name | Initial<br>Value | D/M      | Description   |
|----------|----------|------------------|----------|---|
| 15 to 13 |          | All 0            | R/W<br>R | Description  Reserved   |
| 15 10 13 | _        | All U            | н        |   |
|          |          |                  |          | The write value must always be 0; otherwise operation is not guaranteed.  |
| 12 to 3  | BCADR    | All 0            | R/W      | Blank Check Address Setting Bit   |
|          |          |                  |          | Use these bits to specify the address of the target area when the size of the target area to be checked by the blank check command is 8 bytes (the BCSIZE bit is set to 0). When the BCSIZE bit is set to 0, the start address of the target area is the value obtained by summing the EEPBCCNT value (the value obtained by shifting the set BCADR value by 3 bits) and the start address of an erased block specified when a blank check command is issued. |
| 2, 1     | _        | All 0            | R        | Reserved  |
|          |          |                  |          | The write value must always be 0; otherwise operation is not guaranteed.  |
| 0        | BCSIZE   | 0                | R/W      | Blank Check Size Setting Bit  |
|          |          |                  |          | This bit selects the size of the target area to be checked by the blank check command.  |
|          |          |                  |          | 0: Selects 8 bytes as the size of a blank check target area.  |
|          |          |                  |          | 1: Selects 8 Kbytes as the size of a blank check target area.   |

# 26.3.10 EEPROM Blank Check Status Register (EEPBCSTAT)

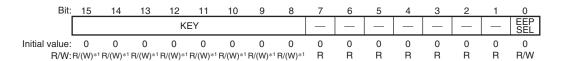
EEPBCSTAT stores check results by executing the blank check command. In on-chip ROM disabled mode, EEPBCSTAT is read as H'0000, and writing to it is ignored. EEPBCSTAT is initialized by a power-on reset, a transition to the hardware standby mode, or by setting the FRESET bit of FRESETR to 1.

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

| Bit     | Bit Name | Initial<br>Value | R/W | Description   |
|---------|----------|------------------|-----|---|
| 15 to 1 | _        | All 0            | R   | Reserved  |
|         |          |                  |     | The write value must be 0; otherwise operation is not guaranteed. |
| 0       | BCST     | 0                | R   | Blank Check Status Bit  |
|         |          |                  |     | Indicates the result of a blank check.                            |
|         |          |                  |     | 0: The target area is erased (blank).                             |
|         |          |                  |     | 1: The target area is filled with 0s and/or 1s.                   |

# 26.3.11 EEPROM MAT Select Register (EEPMAT)

EEPMAT switches memory MATs in the EEPROM. In on-chip ROM disabled mode, EEPMAT is read as H'0000 and writing to it is ignored. EEPMAT is initialized by a power-on reset or a transition to the hardware standby mode.



| Bit     | Bit Name | Initial<br>Value | R/W     | Description  |
|---------|----------|------------------|---------|--|
| 15 to 8 | KEY      | H'00             | R/(W)*1 | Key Code   |
|         |          |                  |         | These bits enable or disable EEPSEL bit modification. The data written to these bits are not stored.   |
| 7 to 1  | _        | All 0            | R       | Reserved   |
|         |          |                  |         | These bits are always read as 0. The write value should always be 0.   |
| 0       | EEPSEL   | 0                | R/W     | EEPROM MAT Select  |
|         |          |                  |         | Selects a memory MAT in the EEPROM. Writing to this bit is enabled only when this register is accessed in word size and H'B3 is written to the KEY bits. |
|         |          |                  |         | 0: Selects the data MAT  |
|         |          |                  |         | 1: Selects the product information MAT* <sup>2</sup>   |

Notes: 1. The data written to are not retained.

2. Product information MAT is read only and no write/erase can be performed.

#### 26.4 Overview of EEPROM-Related Modes

Figure 26.4 shows the EEPROM-related mode transition in this LSI. For the relationship between the LSI operating modes and the MD4 to MD0 and FWE pin settings, refer to section 3, Operating Modes.

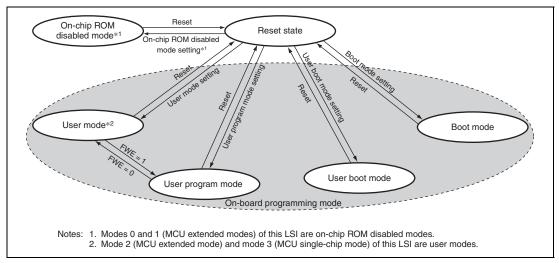


Figure 26.4 EEPROM-Related Mode Transition

- The EEPROM cannot be read, programmed, or erased in on-chip ROM disabled mode.
- The data MAT can be read, programmed, and erased on the board in user mode, user program mode, user boot mode, and boot mode.
- In user mode, the ROM cannot be programmed or erased but the EEPROM can be
  programmed and erased. While the EEPROM is being programmed or erased, the ROM can be
  read. Therefore, the user can program the EEPROM while executing an application program in
  the ROM protected against programming and erasure.

Table 26.3 compares programming- and erasure-related items for the boot mode, user mode, user program mode, and user boot mode.

**Table 26.3 Comparison of Programming Modes** 

| Item                            | Boot Mode                         | User Mode               | User Program<br>Mode    | User Boot Mode          |
|---------------------------------|-----------------------------------|-------------------------|-------------------------|-------------------------|
| Programming/erasure enabled MAT | Data MAT                          | Data MAT                | Data MAT                | Data MAT                |
| Programming/erasure control     | Host                              | FCU                     | FCU                     | FCU                     |
| Programming data transfer       | From host via<br>SCI              | From any device via RAM | From any device via RAM | From any device via RAM |
| Reset-start MAT                 | Embedded<br>program stored<br>MAT | User MAT                | User MAT                | User boot MAT*          |

Note: \* After the LSI is started in the embedded program stored MAT and the embedded program built in the product is executed, execution starts from the location indicated by the reset vector of the user boot MAT.

- In user boot mode, a boot operation with a desired interface can be implemented through mode pin settings different from those in user mode or user program mode.
- Both in boot mode and user boot mode, the embedded program built in the product uses H'FFF88000 to H'FFF8FFFF in the internal RAM. Therefore, once the RAM is disabled via the RAM Enable Control register (RAMEN) and a reset is issued, the data stored in the corresponding area in the RAM prior to the reset is no longer stored in the RAM after booting is initiated in boot mode or user boot mode (see section 28, RAM).

#### 26.5 Boot Mode

To program or erase the data MAT in boot mode, send control commands and programming data from the host. For the system configuration and settings in boot mode, refer to section 25, ROM. This section describes only the commands dedicated for the EEPROM.

## 26.5.1 Inquiry/Selection Host Commands

Table 26.4 shows the inquiry/selection host commands dedicated to the EEPROM. The data MAT inquiry and data MAT information inquiry commands are used in the step for inquiry regarding the MAT programming information shown in figure 25.9 in section 25.5.4, Inquiry/Selection Host Command Wait State.

**Table 26.4** Inquiry/Selection Host Commands (for EEPROM only)

| Host Command Name            | Function   |
|------------------------------|--|
| Data MAT inquiry             | Inquires regarding the availability of user MAT                            |
| Data MAT information inquiry | Inquires regarding the number of data MATs and the start and end addresses |

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

## (1) Data MAT Inquiry

In response to a data MAT inquiry command sent from the host, this LSI returns the information concerning the availability of data MATs.

Command H'2A

Response H'3A Size Availability SUM

[Legend]

Size (1 byte): Total number of characters in the availability field (fixed at 1)

Availability (1 byte): Availability of data MATs (fixed at H'01)

H'00: No data MAT is available H'01: Data MAT is available

SUM (1 byte): Checksum

## (2) Data MAT Information Inquiry

In response to a data MAT information inquiry command sent from the host, this LSI returns the number of data MATs and their addresses.

| Command  | H'2B              |      |           |  |  |  |  |  |  |  |
|----------|-------------------|------|-----------|--|--|--|--|--|--|--|
| Response | H'3B              | Size | MAT count |  |  |  |  |  |  |  |
|          | MAT start address |      |           |  |  |  |  |  |  |  |
|          | MAT end address   |      |           |  |  |  |  |  |  |  |
|          | MAT start address |      |           |  |  |  |  |  |  |  |
|          | MAT end address   |      |           |  |  |  |  |  |  |  |
|          | :                 |      |           |  |  |  |  |  |  |  |
|          | MAT start address |      |           |  |  |  |  |  |  |  |
|          | MAT end address   |      |           |  |  |  |  |  |  |  |
|          | SUM               |      |           |  |  |  |  |  |  |  |

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end

address fields

MAT count (1 byte): Number of data MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of a data MAT MAT end address (4 bytes): End address of a data MAT

SUM (1 byte): Checksum

The information concerning the block configuration in the data MAT is included in the response to the erasure block information inquiry command (refer to section 25.5.4, Inquiry/Selection Host Command Wait State).

### 26.5.2 Programming/Erasing Host Commands

Table 26.5 shows the programming/erasing host commands dedicated to the EEPROM. EEPROM-dedicated host commands are provided only for checksum and blank check; the programming, erasing, and reading commands are used in common for the ROM and EEPROM.

To program the data MAT, issue from the host a user MAT programming selection command and then a 256-byte programming command specifying a data MAT address as the programming address. To erase the data MAT, issue an erasure selection command and then a block erasure command specifying an erasure block in the data MAT. The information concerning the erasure block configuration in the data MAT is included in the response to the erasure block information inquiry command. To read data from the data MAT, select the user MAT through a memory read command specifying a data MAT address as the read address.

For the user MAT programming selection, user boot MAT programming selection, 256-byte programming, erasure selection, block erasure selection, and memory read commands, refer to section 25.5.5, Programming/Erasing Host Command Wait State. For the erasure block information inquiry command, refer to section 25.5.4, Inquiry/Selection Host Command Wait State.

 Table 26.5
 Programming/Erasure Host Commands (for EEPROM)

| Host Command Name    | Function  |
|----------------------|---|
| Data MAT checksum    | Performs checksum verification for the data MAT |
| Data MAT blank check | Checks whether the data MAT is blank            |

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

### (1) Data MAT Checksum

In response to a data MAT checksum command sent from the host, this LSI sums the data MAT data in byte units and returns the result (checksum).

Command

H'61

Response

H'71 Size

MAT checksum

SUM

[Legend]

Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4)

MAT checksum (4 bytes): Checksum of the data MAT data

SUM (1 byte): Checksum (for the response data)

#### (2) Data MAT Blank Check

In response to a data MAT blank check command sent from the host, this LSI checks whether the data MAT is completely erased. When the data MAT is completely erased, this LSI returns a response (H'06). If the user MAT has an unerased area, this LSI returns an error response (sends H'E2 and H'52 in that order).

Command

H'62

Response

H'06

Error response

H'E2 H'52

# 26.6 User Mode, User Program Mode, and User Boot Mode

#### 26.6.1 FCU Command List

To program or erase the data MAT in user mode, user program mode, or user boot mode, issue FCU commands to the FCU. Table 26.6 is a list of FCU commands for EEPROM programming and erasure

**Table 26.6 FCU Command List (EEPROM-Related Commands)** 

| Command   | Function   |
|---|--|
| Normal mode transition                          | Moves to the normal mode (see section 26.6.2, Conditions for FCU Command Acceptance).        |
| Status read mode transition                     | Moves to the status read mode (see section 26.6.2, Conditions for FCU Command Acceptance).   |
| Lock bit read mode transition (lock bit read 1) | Moves to the lock bit read mode (see section 26.6.2, Conditions for FCU Command Acceptance). |
| Program   | Programs EEPROM (in 8-byte or 128-byte units).   |
| Block erase                                     | Erases EEPROM (in block units).  |
| P/E suspend                                     | Suspends programming or erasure.   |
| P/E resume                                      | Resumes programming or erasure.  |
| Status register clear                           | Clears the IRGERR, ERSERR, and PRGERR bits in FSTATR0 and cancels the command-locked state.  |
| Blank check                                     | Checks if a specified area is erased (blank).  |

FCU commands other than the program command and blank check command are also used for ROM programming and erasure. When the blank check command is issued to the ROM, the lock bits in the ROM are read out.

To issue a command to the FCU, access the EEPROM area through the P bus. Table 26.7 shows the FCU command formats for the program command and blank check command. For the other command formats, refer to section 25.6.1, FCU Command List. When a P-bus access, as shown in table 26.7, is made under specified conditions, the FCU performs processing specified by a selected command. For the conditions for the FCU command acceptance, refer to section 26.6.2, Conditions for FCU Command Acceptance. For details of command usage, refer to section 26.6.3, FCU Command Usage.

When the FRDMD bit is set to 0 (memory area read mode), if the data in the first cycle of an FCU command is determined as H'71, the FCU accepts the lock bit read mode transition command. Since the EEPROM has no lock bits, making P-bus access after a transition to the lock bit read mode results in undefined read data. The FCU detects no access violation error when the undefined data is read. When the FRDMD bit is set to 1 (register read mode), if the data in the first cycle of an FCU command is determined as H'71, the FCU enters a waiting state to wait for the command in the second cycle (H'D0) of the blank check command. At this stage, if H'D0 is written into an EEPROM area by a P-bus write access, the FCU detects it and starts performing the blank check processes specified by the set values in the EEPBCCNT register, and once the check completes the FCU writes check results into the EEPBCSTAT register.

There are two suspending modes to be initiated by the P/E suspend command; the suspension-priority mode and erasure-priority mode. For details of each mode, refer to section 25.6.4, Suspending Operation.

**Table 26.7 FCU Command Formats (for EEPROM only)** 

|  | Number           | First C | ycle | Second  | Cycle | Third C | ycle | Fourth C | •    | Cycle N + 3 |      |
|--|------------------|---------|------|---------|-------|---------|------|----------|------|-------------|------|
| Command                                | of Bus<br>Cycles | Address | Data | Address | Data  | Address | Data | Address  | Data | Address     | Data |
| Program (8-byte programming: N = 4)    | 7                | EA      | H'E8 | EA      | H'04  | WA      | WD1  | EA       | WDn  | EA          | H'D0 |
| Program (128-byte programming: N = 64) | 67               | EA      | H'E8 | EA      | H'40  | WA      | WD1  | EA       | WDn  | EA          | H'D0 |
| Blank check                            | 2                | EA      | H'71 | ВА      | H'D0  | _       | _    | _        | _    | _           | _    |

#### [Legend]

EA: EEPROM area address

An arbitrary address within the range of H'8010000 to H'8011FFFF

WA: The start address of write data

BA: The address of an EEPROM erasure block

(An arbitrary address in the erase target block)

WDn: n-th word of programming data (n = 1 to N)

# 26.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 26.5 is an FCU mode transition diagram.

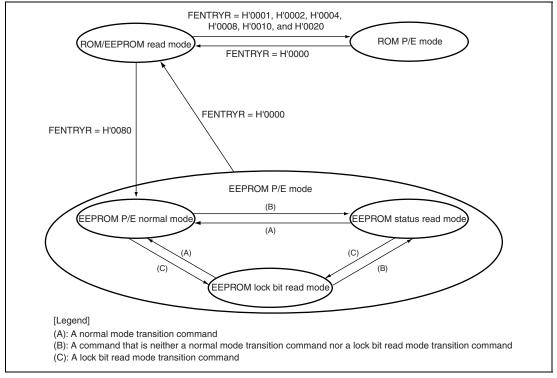


Figure 26.5 FCU Mode Transition Diagram (EEPROM-Related Modes)

### (1) ROM P/E Mode

The FCU can accept ROM programming and erasing commands in this mode. The EEPROM cannot be read. The FCU enters this mode when the FENTRYD bit is set to 0 and any one of the bits, FENTRY5 to FENTRY0, is set to 1 in FENTRYR. For details of this mode, refer to section 25.6.2, Conditions for FCU Command Acceptance.

### (2) ROM/EEPROM Read Mode

The EEPROM can be read through the peripheral bus, and the ROM can be read through the ROM cache at a high speed. The FCU does not accept commands. The FCU enters this mode when the FENTRY5 to FENTRY9 bits are set to 000000 and the FENTRYD bit to 0 in FENTRYR.

#### (3) EEPROM P/E Mode

#### EEPROM P/E normal mode

The FCU enters this mode when the FENTRYD bit is set to 1 and the FENTRY5 to FENTRY0 bits are set to 000000 in ROM/EEPROM read mode or ROM P/E mode, or when a normal mode transition command is accepted in EEPROM P/E mode. Table 26.8 shows the commands that can be accepted in this mode. If the EEPROM area is read through the P bus, an EEPROM access error occurs and the FCU enters the command-locked state.

#### EEPROM status read mode

The FCU enters this mode when the FCU accepts a command that is neither the normal mode transition command nor the lock bit read mode transition command in EEPROM P/E mode. The EEPROM status read mode includes the state in which the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 26.8 shows the commands that can be accepted in this mode. If the EEPROM area is read through the P bus, the FSTATR0 value is read.

#### EEPROM lock bit read mode

The FCU enters this mode when the FCU accepts a lock bit read mode transition command in EEPROM P/E mode. Table 26.8 shows the commands that can be accepted in this mode. Since the EEPROM has no lock bits, reading an EEPROM area via the P-bus results in an undefined value. However, no access violation occurs in this case.

Table 26.8 shows the correlation between each FCU mode and its register /state and its acceptable commands. When an unacceptable command is issued, the FCU enters the command-locked state (see section 26.7.3, Error Protection).

To make sure that the FCU accepts a command, enter the mode in which the FCU can accept the target command, check the FRDY, ILGLERR, ERSERR, and PRGERR bit values in FSTATR0, and the FCUERR, FRDTCT, and FRCRCT bit values in FSTATR1, and then issue the target FCU command. The CMDLK bit in FASTAT holds a value obtained by logical ORing the ILGLERR, ERSERR, and PRGERR bit values in FSTATR0 and the FCUERR, FRDTCT, and FRCRCT bit values in the FSTATR1. Therefore the FCU's error occurrence state can be checked by reading the CMDLK bit. In table 26.8, the CMDLK bit is used as the bit to indicate the error occurrence state. The FRDY bit of FSTATR0 is 0 during the programming/erasure, programming/erasure suspension, and blank check processes. While the FRDY bit is 0, the P/E suspend command can be accepted only when the SUSRDY bit in FSTATR0 is 1.

Table 26.8 includes 0 and 1 in single cells of the ERSSPD, PRGSPD, and FRDY bit rows for the sake of simplification. The ERSSPD bits 1 and 0 indicate the erasure suspension and programming suspension processes, respectively. The PRGSPD bits 1 and 0 indicate the programming suspension and erasure suspension processes, respectively. The FRDY bit value can be either 1 or 0, which is a value held by the bit prior to a transition to the command lock state.

Table 26.8 FCU Modes/States and Acceptable Commands

|   | P/E I<br>Mod              |                   | mal         | Stat                              | Status Read Mode                                      |  |                           |                           |                   |                           | Lock Bit<br>Read Mode        |             |                           |                   |             |
|---|---------------------------|-------------------|-------------|-----------------------------------|---|--|---------------------------|---------------------------|-------------------|---------------------------|------------------------------|-------------|---------------------------|-------------------|-------------|
| ltem  | Programming-<br>Suspended | Erasure-Suspended | Other State | Programming/Erasure<br>Processing | Programming<br>Processing during<br>Erasure-Suspended | Programming/Erasure<br>Suspension Processing | Blank Check<br>Processing | Programming-<br>Suspended | Erasure-Suspended | Command-Locked (FRDY = 0) | Command-Locked<br>(FRDY = 1) | Other State | Programming-<br>Suspended | Erasure-Suspended | Other State |
| FRDY bit in FSTATR0                             | 1                         | 1                 | 1           | 0                                 | 0   | 0  | 0                         | 1                         | 1                 | 0                         | 1                            | 1           | 1                         | 1                 | 1           |
| SUSRDY bit in FSTATR0                           | 0                         | 0                 | 0           | 1                                 | 0   | 0  | 0                         | 0                         | 0                 | 0                         | 0                            | 0           | 0                         | 0                 | 0           |
| ERSSPD bit in FSTATR0                           | 0                         | 1                 | 0           | 0                                 | 1   | 0/1  | 0/1                       | 0                         | 1                 | 0/1                       | 0/1                          | 0           | 0                         | 1                 | 0           |
| PRGSPD bit in FSTATR0                           | 1                         | 0                 | 0           | 0                                 | 0   | 0/1  | 0/1                       | 1                         | 0                 | 0/1                       | 0/1                          | 0           | 1                         | 0                 | 0           |
| CMDLK bit in FASTAT                             | 0                         | 0                 | 0           | 0                                 | 0   | 0  | 0                         | 0                         | 0                 | 1                         | 1                            | 0           | 0                         | 0                 | 0           |
| Normal mode transition                          | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                         | Α                         | Α                 | ×                         | ×                            | Α           | Α                         | Α                 | Α           |
| Status read mode transition                     | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                         | Α                         | Α                 | ×                         | ×                            | Α           | Α                         | Α                 | Α           |
| Lock bit read mode transition (lock bit read 1) | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                         | Α                         | Α                 | ×                         | ×                            | Α           | Α                         | Α                 | Α           |
| Program   | ×                         | *                 | Α           | ×                                 | ×   | ×  | ×                         | ×                         | *                 | ×                         | ×                            | Α           | ×                         | *                 | Α           |
| Block erase                                     | ×                         | ×                 | Α           | ×                                 | ×   | ×  | ×                         | ×                         | ×                 | ×                         | ×                            | Α           | ×                         | ×                 | Α           |
| P/E suspend                                     | ×                         | ×                 | ×           | Α                                 | ×   | ×  | ×                         | ×                         | ×                 | ×                         | ×                            | ×           | ×                         | ×                 | ×           |
| P/E resume                                      | Α                         | Α                 | ×           | ×                                 | ×   | ×  | ×                         | Α                         | Α                 | ×                         | ×                            | ×           | Α                         | Α                 | ×           |
| Status register clear                           | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                         | Α                         | Α                 | ×                         | Α                            | Α           | Α                         | Α                 | Α           |
| Blank check                                     | Α                         | Α                 | Α           | ×                                 | ×   | ×  | ×                         | Α                         | Α                 | ×                         | ×                            | Α           | Α                         | Α                 | Α           |

[Legend]

A: Acceptable

<sup>\*:</sup> Only programming is acceptable for the areas other than the erasure-suspended block

<sup>×:</sup> Not acceptable

#### 26.6.3 FCU Command Usage

This section shows how to program and erase the EEPROM using the program command and block erase command, respectively, and how to check the erasure status of the EEPROM using the blank check command. For the firmware transfer to the FCU RAM and the other FCU command usage, refer to section 25.6.3, FCU Command Usage.

If the FCU enters the command lock state in the middle of its handling of commands by setting the FCUERR, FRDTCT, or FRCRCT bit in FSTATR1 to 1, the FRDY bit in FSTATR0 retains 0. Since the FCU halts its operation in the command lock state, the FRDY bit is not set to 1 from 0.

If the FRDY bit retains 0 for longer than the programming/erasure time or suspend delay time (see section 32, Electrical Characteristics), an abnormal operation may have occurred. In such case, initialize the FCU by issuing an FCU reset.

If the FRDY bit is set to 1 upon the termination of an FCU command operation, both the FCUERR, FRDTCT, and FRCRCT bits are cleared to 0. On the other hand, it can be checked via the ILGLERR, ERSERR, or PRGERR bit whether or not an error has occurred after a command operation terminates.

### (1) Programming

To program the EEPROM, use the program command. Write byte H'E8 to an EEPROM area address in the first cycle of the program command and the number of words (N)\* to be programmed through byte access in the second cycle. Access the P bus in words from the third cycle to cycle N + 2 of the command. In the third cycle, write the programming data to the start address of the target programming area. Here, the start address must be an 8-byte boundary address for 8-byte programming or a 128-byte boundary address for 128-byte programming. After writing words to EEPROM area addresses N times, write byte H'D0 to an EEPROM area address in cycle N + 3; the FCU then starts EEPROM programming. Read the FRDY bit in FSTATR0 to confirm that EEPROM programming is completed.

If the area accessed in the third cycle to cycle N + 2 includes addresses that do not need to be programmed, write H'FFFF as the programming data for those addresses. To ignore the programming and erasure protection provided by the EEPWE0 and EEPWE1 settings, set the program/erase enable bit for the target block to 1 before starting programming. To ignore the protection provided by the lock bit during programming, set the FPROTCN bit in FPROTR to 1 before starting programming. Figure 26.6 shows the procedure for EEPROM programming

Note: \* N = H'04 for 8-byte programming or N = H'40 for 128-byte programming.

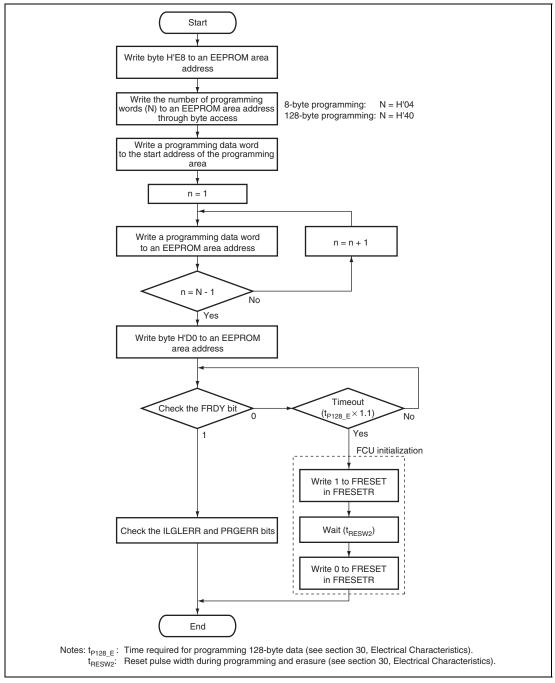


Figure 26.6 Procedure for EEPROM Programming

#### (2) Erasure

To erase the ROM, use the block erase command. The EEPROM can be erased in the same way as ROM erasure (refer to section 25, ROM). Note that the EEPROM has a programming and erasure protection function through EEPWE0 and EEPWE1. To ignore the programming and erasure protection provided by the EEPWE0 and EEPWE1 settings, set the program/erase enable bit for the target block to 1 before starting erasure.

### (3) Checking of the Erased State

Since reading the EEPROM erased by the CPU results in undefined values, the blank check command should be used to check the erased state of the EEPROM. To make the blank check command available for use, set the FRDMD bit in FMODR to 1 to enable the command first, and then specify the size and start address of a target area via the EEPBCCNT register. When the BCSIZE bit of the EEPBCCNT register is set to 1, a check can be performed on the entire erased block (8 Kbytes) specified in the second cycle of the command. When the BCSIZE bit is set to 0, a check can be performed on an 8-byte area starting from the address obtained by summing the start address of the erased area specified in the second cycle of the command and the value held by the EEPBCCNT register. In the first cycle of the command, a value of H'71 is written in byte into an address of the EEPROM. In the second cycle, once a value of H'D0 is written into a specified address included in the target area, the FCU starts the blank check on the EEPROM. It can be checked whether or not the check is complete via the FRDY bit in the FSTATRO. After the blank check is complete, it can be checked whether the target area is erased or filled with 0s and/or 1s via the BCST bit of the EEPBCSTAT register.

Figure 26.7 shows the procedure of the EEPROM blank check.

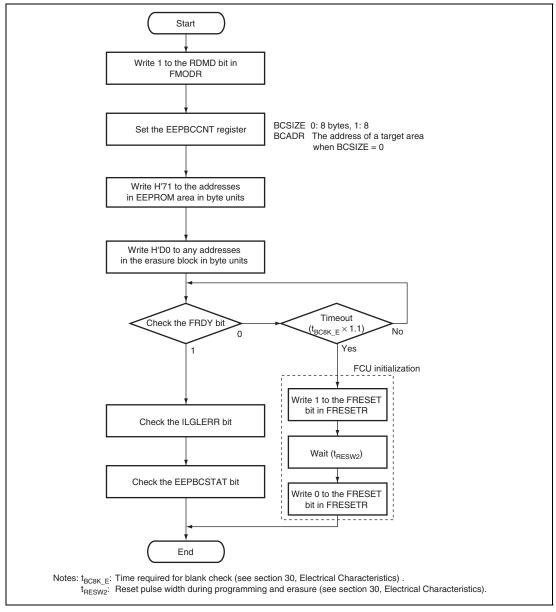


Figure 26.7 Procedure of the EEPROM Blank Check

### 26.7 Protection

There are three types of EEPROM programming/erasure protection: hardware, software, and error protection.

#### 26.7.1 Hardware Protection

The hardware protection function disables EEPROM programming and erasure according to the mode pin settings in this LSI.

While the on-chip ROM is disabled, EEPROM programming, erasing, and reading are disabled. For the operating modes set through the mode pins of this LSI, refer to section 3, Operating Modes.

#### 26.7.2 Software Protection

The software protection function disables EEPROM programming and erasure according to the control register settings. If an attempt is made to issue a programming or erasing command to the EEPROM against software protection, the FCU detects an error and enters command-locked state.

### (1) Protection through FENTRYR

When the FENTRYD bit in FENTRYR is 0, the FCU does not accept commands for the EEPROM, so EEPROM programming and erasure are disabled. If an attempt is made to issue an FCU command for the EEPROM while the FENTRYD bit is 0, the FCU detects an illegal command error and enters command-locked state (see section 26.7.3, Error Protection).

# (2) Protection through EEPWE0 and EEPWE1

When the DBWEi (i = 00 to 15) bit in EEPWE0 or EEPWE1 is 0, programming and erasure of block DBi in the data MAT is disabled. If an attempt is made to program or erasure block DBi while the DBWEi bit is 0, the FCU detects a program/erase protect error and enters command-locked state (see section 26.7.3, Error Protection).

#### 26.7.3 Error Protection

The error protection function detects an illegal FCU command issued, an illegal access, or an FCU malfunction, and disables FCU command acceptance (command-locked state). While the FCU is in command-locked state, the EEPROM cannot be programmed or erased. To cancel command-locked state, issue a status register clear command while FASTAT is H'10.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFE) interrupt is generated if the FCU enters command-locked state (the CMDLK bit in FASTAT becomes 1). While an EEPROM-related interrupt enable bit (EEPAEIE, EEPIFEIE, EEPRPEIE, or EEPWPEIE) in FAEINT is 1, an FIFE interrupt is generated if the corresponding status bit (EEPAE, EEPIFE, EEPRPE, or EEPWPE) in FASTAT becomes 1.

Table 26.9 shows the error protection types for the EEPROM and the status bit values (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the EEPAE, EEPIFE, EEPRPE, and EEPWPE bits in FASTAT) after each error detection. For the error protection types used in common by the ROM and EEPROM (FENTRYR setting error, most of illegal command errors, erasing error, programming error, FCU error, and FCU RAM ECC error), refer to section 25.8.3, Error Protection. If the FCU enters command-locked state due to a command other than a suspend command issued during programming or erasure processing, the FCU continues programming or erasing the EEPROM. In this state, the P/E suspend command cannot suspend programming or erasure. If a command is issued in command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to the previous error detection.

**Table 26.9 Error Protection Types (for EEPROM only)** 

| Error                                    | Description  | ILGLERR | ERSERR | PRGERR | EEPAE | EEPIFE | EEPRPE | EEPWPE |
|--|--|---------|--------|--------|-------|--------|--------|--------|
| Illegal<br>command                       | The value specified in the second cycle of a program command is neither H'04 nor H'40.   | 1       | 0      | 0      | 0     | 0      | 0      | 0      |
| error                                    | A lock bit program command has been issued to an area in the EEPROM while the FENTRYD bit of FENTRYR register is set to 1.                       | 1       | 0      | 0      | 0     | 0      | 0      | 0      |
| EEPROM access error                      | A read access command has been issued to the EEPROM area while FENTRYD = 1 in FENTRYR in EEPROM P/E normal mode.                                 | 1       | 0      | 0      | 1     | 0      | 0      | 0      |
|  | A write access command has been issued to the EEPROM area while FENTRYD = 0.   | 1       | 0      | 0      | 1     | 0      | 0      | 0      |
|  | An access command has been issued to the EEPROM area while one of the bits, FENTRY5 to FENTRY0, in FENTRYR is 1.                                 | 1       | 0      | 0      | 1     | 0      | 0      | 0      |
| EEPROM<br>instruction<br>fetch error     | An instruction fetch has been made in the EEPROM area.   | 1       | 0      | 0      | 0     | 1      | 0      | 0      |
| EEPROM read protect error                | A read access command has been issued to the EEPROM area protected against reading through EEPRE0 and EEPRE1.                                    | 1       | 0      | 0      | 0     | 0      | 1      | 0      |
| EEPROM<br>program/erase<br>protect error | A program command or block erase command has been issued to the EEPROM area protected against programming and erasure through EEPWE0 and EEPWE1. | 1       | 0      | 0      | 0     | 0      | 0      | 1      |

## 26.8 Product Information MAT

The product information MAT stores the device name, device revision number, and embedded program revision number information in ASCII code. The embedded program is stored in the reset-start MAT used in boot mode and user boot mode (refer to section 25.4, Overview of ROM-Related Modes). Table 26.10 shows the addresses to store the information and an example of information data. In the product information MAT (H'80100000 to H'8010007F), the addresses not shown in this table are reserved areas. Undefined data will be read from the reserved areas.

Table 26.10 Data Stored in Product Information MAT

| Information                      | Address                  | Example of Data                                    |
|----------------------------------|--------------------------|--|
| Device name                      | H'80100000 to H'8010000F | H'523546373235343652202020<br>20202020 = R5F72546R |
| Device revision number           | H'80100010 to H'80100011 | H'3031 = 01  |
| Embedded program revision number | H'80100020 to H'80100022 | H'313030 = 100 (1.00)                              |

# 26.9 Usage Notes

### (1) Protection of Data MAT Immediately after a Reset

As the initial value of EEPRE0, EEPRE1, EEPWE0, and EEPWE1 is H'0000, data MAT programming, erasure, and reading are disabled immediately after a reset. To read data from the data MAT, set EEPRE0 and EEPRE1 appropriately before accessing the data MAT. To program or erase the data MAT, set EEPWE0 and EEPWE1 appropriately before issuing an FCU command for programming or erasure. If an attempt is made to read, program, or erase the data MAT without setting the registers, the FCU detects an error and enters command-locked state.

### (2) State in which AUD Operation Is Disabled and Interrupts Are Ignored

In the following modes or period, the AUD is in module standby mode and cannot operate. The NMI or maskable interrupt requests are ignored.

- Boot mode
- The program in the embedded program stored MAT is being executed immediately after the LSI is started in user boot mode

### (3) Programming-/Erasure-Suspended Area

The data stored in the programming-suspended or erasure-suspended area is undetermined. To avoid malfunction due to undefined read data, ensure that no data is read from the programming-suspended or erasure-suspended area.

## (4) Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcontrollers

The flash memory programming/erasing program used for conventional F-ZTAT SH microcontrollers does not work with this LSI.

SH72546R Section 26 EEPROM

### (5) Reset during Programming or Erasure

To reset the FCU by setting the FRESET bit in the FRESETR register during programming or erasure, hold the FCU in the reset state for a period of t<sub>RESW2</sub> (see section 32, Electrical Characteristics). Since a high voltage is applied to the EEPROM during programming and erasure, the FCU has to be held in the reset state long enough to ensure that the voltage applied to the memory unit has dropped. Do not read from the EEPROM while the FCU is in the reset state.

When a power-on reset is generated by asserting the  $\overline{RES}$  pin during programming or erasure of the flash memory, hold the reset state for a period of  $t_{RESW2}$  (see section 32, Electrical Characteristics). In a power-on reset, not only does the voltage applied to the memory unit have to drop, but the power supply for the EEPROM and its internal circuitry also have to be initialized. Thus, the reset state must be maintained over a longer period than in the case of resetting the FCU.

While programming or erasure is performed, do not generate an internal reset caused by WDT counter overflow. A reset caused by WDT cannot ensure a sufficient time required for voltage drop for the memory unit, initialization of the power supply for the EEPROM, or initialization of its internal circuit.

When either a power-on reset by asserting the RES pin, or an FCU reset by setting the FRESET bit in the FRESETR register, is executed during programming or erasure, the whole data in the programming or erasure area becomes undefined.

No verification function is provided to check program/erase state of the area where the data is undefined by suspend of program/erase (e.g., reset input, power-supply interruption). Therefore, if the undefined area should be used again, make sure to completely erase data before usage.

## (6) Prohibition of Additional Programming

One area cannot be programmed twice in succession. To program an area that has already been programmed, be sure to erase the area before reprogramming.

## (7) Writing to/Erasing the Product Information MAT

Since the product information MAT is read only, no write/erase operation can be performed. When write/erase operation is performed with the EEPSEL bit in the EEPMAT register set to 1, the data MAT is written to or erased without any error occurring such as EEPROM access violation. Therefore, never write to or erase the product MAT.

Section 26 EEPROM SH72546R

### (8) Suspension by Programming or Erasure Suspension

When programming or erasure is suspended by a programming or an erasure suspend command, the programming or erasure must be completed by a resume command.

### (9) Power off during Programming or Erasure

Do not switch the power off during programming or erasure. As a high voltage is applied to the ROM during programming and erasure, a certain period is required for this voltage to fall.

Thus, to allow for cases where switching the power off during programming or erasure cannot be avoided, design the system so that at least the  $V_{cc}$  holding time at  $PV_{cc}$  shutdown ( $t_{vcch}$ ) is secured by assertion of the  $\overline{HSTBY}$  signal causing a transition to the hardware standby state if the power is switched off. For details, see section 32, Electrical Characteristics.

### (10) Prohibition of Clearing FRDCLE Bit to 0

Whenever the FRDTCT bit in FSTATR1 is set to 1, the FCU has to enter command-locked state, because the FCU command operation cannot be guaranteed. Thus, do not clear the FRDCLE bit in FRAMECCR to 0.

# Section 27 ROM Cache (ROMC)

The ROM cache is designed to cache the instructions and data stored in the ROM, permitting high-speed access to these instructions and data.

#### 27.1 Features

- Configuration: Separate caches for instructions and data
- Prefetch cache: 8-line, 4-way set associative, LRU method
- Prefetch-miss cache: 4-line, fully associative, LRU method
- Data cache: 4-line, fully associative, LRU method, automatic line invalidation
- Line size: 16 bytes (128 bits)
- Hardware prefetching: Instructions are read from the ROM and stored in the prefetch cache prior to instruction fetching by the CPU

## 27.2 Configuration

The ROM-cache module has separate units for instructions and data. The instruction caches consist of a prefetch cache (for instruction read-ahead) and prefetch-miss cache (where data read from the ROM are stored when neither the prefetch cache nor the prefetch-miss cache currently contains the desired data). The data cache is for the storage of data.

Figure 27.1 shows the configuration of the caches. The prefetch cache is 8-line and 4-way set associative, while both the prefetch-miss cache and the data cache are 4-line full associative. All of the caches have 16-byte lines, and the LRU (Least Recently Used) method is the principle of the line-replacement algorithm. For detailed descriptions of the line-replacement policies for the individual caches, refer to section 27.4.1, Data Cache Lookup.

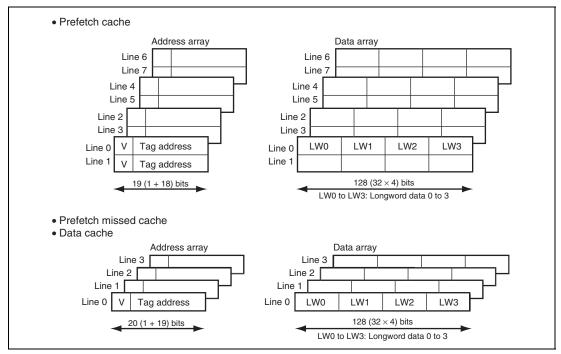


Figure 27.1 Cache Configuration

#### (1) Address Arrays

The address array of the prefetch cache consists of eight lines, lines 0 to 7. The address arrays of the data and prefetch-miss caches consist of four lines, lines 0 to 3. Each line is composed of a V bit and a Tag address.

The V bit indicates whether or not the line data is valid; the data line is valid when the V bit is 1 and invalid when the V bit is 0. When the target address in the ROM is identified as already cached, the corresponding line is invalidated by setting the V bit to 0.

The Tag addresses hold the addresses for reference in the cache lookup process. Each tag consists of 18 bits (corresponding to bits 22 to 5 of the access address in the ROM) in the prefetch cache and 19 bits (bits 22 to 4 of the access address) in the prefetch-miss cache and the data cache. Bits 31 to 23 of the access address are used on the address bus to identify the memory space, and are thus not relevant to the cache lookup process.

The V bits for the instruction caches or data cache are initialized to 0 by writing a 1 to the corresponding flush bit in the ROM cache control register. The whole cache is initialized to 0 by writing a 1 to the main flush bit in the ROM cache control register and on entry to the reset or standby state.

#### (2) Data Array

The data arrays of the data and prefetch-miss caches consist of four lines, lines 0 to 3. Each line is 16 bytes long and is divided into four longwords, LW0 to LW3. The 16 bytes of instructions or data held by a single line is the smallest unit of caching.

Data stored in the data array are undefined after a reset.

## 27.3 Register Descriptions

The ROM cache has the following registers. These registers can only be accessed as longwords. Table 27.1 shows the configuration of the cache-related registers.

**Table 27.1 Register Configuration** 

| Register Name                | Abbreviation | R/W | Initial Value | Address     | Access Size |
|------------------------------|--------------|-----|---------------|-------------|-------------|
| ROM cache control register   | RCCR         | R/W | H'0000 0001   | H'FFFC 1400 | 32          |
| ROM cache control register 2 | RCCR2        | R/W | H'0000 00F5   | H'FFFC 1408 | 32          |

### 27.3.1 ROM Cache Control Register (RCCR)

The RCE bit of RCCR specifies enabling or disabling of the ROM cache. RCCR also contains the RCF bit, which can be used to invalidate all lines in the ROM cache, the RCFI bit, which can be used to invalidate all lines of the instruction caches (the prefetch and prefetch-miss caches), the RCFD bit, which can be used to invalidate all lines in the data cache.

| Bit:           | 31   | 30      | 29 | 28 | 27 | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19   | 18    | 17     | 16  |
|----------------|--|---------|----|----|----|----|----|----|----------|----|----|----|------|-------|--------|-----|
|                | -  | -       | -  | 1  | -  | 1  | -  | 1  | -        | -  | 1  | -  | -    | -     | -      | -   |
| Initial value: | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0    | 0     | 0      | 0   |
| R/W:           | R  | R       | R  | R  | R  | R  | R  | R  | R        | R  | R  | R  | R    | R     | R      | R   |
| Bit:           | 15   | 14      | 13 | 12 | 11 | 10 | 9  | 8  | 7        | 6  | 5  | 4  | 3    | 2     | 1      | 0   |
| Dit.           | <u> Г.                                    </u> | <u></u> |    |    |    | -  |    |    | <u> </u> |    |    | Ė  | RCF  |       | RCFD   | RCE |
|                |  |         |    |    |    |    |    |    |          |    |    |    | 1101 | 11011 | 1101 D | HOL |
| Initial value: | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0    | 0     | 0      | 1   |
| R/W:           | R  | R       | R  | R  | R  | R  | R  | R  | R        | R  | R  | R  | R/W  | R/W   | R/W    | R/W |

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 31 to 4 | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 3       | RCF      | 0                | R/W | ROM Cache Flush  |
|         |          |                  |     | Writing a 1 to this bit clears the V bits of all lines in the ROM cache to 0 (flushes the cache). This bit is read as 0.   |
|         |          |                  |     | 0: Does not clear the V bits in the ROM cache lines.   |
|         |          |                  |     | 1: Clears the V bits in the ROM cache lines.   |
|         |          |                  |     | [Clearing condition]   |
|         |          |                  |     | Reset/standby  |
|         |          |                  |     | [Setting condition]  |
|         |          |                  |     | Writing a 1.   |
| 2       | RCFI     | 0                | R/W | Instruction Cache Flush  |
|         |          |                  |     | Writing a 1 to this bit clears the V bits of all lines in the prefetch and prefetch-miss-caches to 0 (flushes the prefetch and prefetch-miss caches). This bit is read as 0. |
|         |          |                  |     | 0: Does not clear the V bits in the instruction cache lines.   |
|         |          |                  |     | 1: Clears the V bits in the instruction cache lines.   |
|         |          |                  |     | [Clearing condition]   |
|         |          |                  |     | Reset/standby  |
|         |          |                  |     | [Setting condition]  |
|         |          |                  |     | Writing a 1.   |
| 1       | RCFD     | 0                | R/W | Data Cache Flush   |
|         |          |                  |     | Writing a 1 to this bit clears the V bits of all lines in the data cache to 0 (flushes the data cache). This bit is read as 0.   |
|         |          |                  |     | 0: Does not clear the V bits in the data cache lines.  |
|         |          |                  |     | 1: Clears the V bits in the data cache lines.  |
|         |          |                  |     | [Clearing condition]   |
|         |          |                  |     | Reset/standby  |
|         |          |                  |     | [Setting condition]  |
|         |          |                  |     | Writing a 1.   |
|         |          |                  |     |  |

|     |          | Initial |     |  |
|-----|----------|---------|-----|--|
| Bit | Bit Name | Value   | R/W | Description                                  |
| 0   | RCE      | 1       | R/W | ROM Cache Enable                             |
|     |          |         |     | Specifies usage or non-usage of ROM caching. |
|     |          |         |     | 0: The ROM cache function is not used.       |
|     |          |         |     | 1: The ROM cache function is used.           |
|     |          |         |     | [Clearing condition]                         |
|     |          |         |     | Writing a 0.                                 |
|     |          |         |     | [Setting conditions]                         |
|     |          |         |     | Reset/standby                                |
|     |          |         |     | Writing a 1.                                 |

### 27.3.2 ROM Cache Control Register 2 (RCCR2)

The PCE2 and PCE0 bits of RCCR2 specify enabling or disabling the prefetch-miss cache and data cache, respectively. The PFE bit specifies enabling or disabling the prefetch cache.

With regard to the prefetching function, the PFECF, PFENB, and PFECB bits are used to specify prefetching of consecutive instructions, prefetching from the destinations of unconditional branches, and prefetching from the destinations of conditional branches, respectively.

| Bit:           | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22    | 21    | 20  | 19 | 18   | 17 | 16   |
|----------------|----|----|----|----|----|----|----|----|-------|-------|-------|-----|----|------|----|------|
|                | -  | -  | -  | -  | -  | -  | -  | -  | -     | -     | -     | -   | -  | - 1  | -  | - 1  |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0     | 0   | 0  | 0    | 0  | 0    |
| R/W:           | R  | R  | R  | R  | R  | R  | R  | R  | R     | R     | R     | R   | R  | R    | R  | R    |
| Bit:           | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7     | 6     | 5     | 4   | 3  | 2    | 1  | 0    |
|                | -  | -  | -  | 1  | -  | -  | -  | -  | PFECB | PFENB | PFECF | PFE | -  | PCE2 | -  | PCE0 |
| Initial value: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1     | 1     | 1     | 1   | 0  | 1    | 0  | 1    |

|         |          | Initial |     |  |
|---------|----------|---------|-----|--|
| Bit     | Bit Name | Value   | R/W | Description  |
| 31 to 8 | _        | All 0   | R   | Reserved   |
|         |          |         |     | These bits are always read as 0. The write value should always be 0.                               |
| 7       | PFECB    | 1       | R/W | Conditional Branch Prefetch Enabled  |
|         |          |         |     | Specifies whether or not data is to be prefetched from the destinations of conditional branches.   |
|         |          |         |     | 0: No prefetching from the destinations of conditional branches.                                   |
|         |          |         |     | <ol> <li>Prefetching from the destinations of conditional<br/>branches.</li> </ol>                 |
|         |          |         |     | [Clearing conditions]  |
|         |          |         |     | Reset/standby  |
|         |          |         |     | Writing a 0.   |
|         |          |         |     | [Setting condition]  |
|         |          |         |     | Writing a 1.   |
| 6       | PFENB    | 1       | R/W | Unconditional Branch Prefetch Enable   |
|         |          |         |     | Specifies whether or not data is to be prefetched from the destinations of unconditional branches. |
|         |          |         |     | <ol> <li>No prefetching from the destinations of<br/>unconditional branches.</li> </ol>            |
|         |          |         |     | Prefetching from the destinations of unconditional branches.                                       |
|         |          |         |     | [Clearing conditions]  |
|         |          |         |     | Reset/standby  |
|         |          |         |     | Writing a 0.   |
|         |          |         |     | [Setting condition]  |
| -       |          |         |     | Writing a 1.   |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 5   | PFECF    | 1                | R/W | Consecutive Prefetch Enable   |
|     |          |                  |     | Specifies whether or not prefetching is applied to instructions for consecutive execution (consecutive instructions). |
|     |          |                  |     | 0: Consecutive instructions are not prefetched.   |
|     |          |                  |     | 1: Consecutive instructions are prefetched.   |
|     |          |                  |     | [Clearing conditions]   |
|     |          |                  |     | Reset/standby   |
|     |          |                  |     | Writing a 0.  |
|     |          |                  |     | [Setting condition]   |
|     |          |                  |     | Writing a 1.  |
| 4   | PFE      | 1                | R/W | Prefetch Cache Enable   |
|     |          |                  |     | Specifies usage or non-usage of the prefetch cache.   |
|     |          |                  |     | 0: The prefetch cache function is not used.   |
|     |          |                  |     | 1: The prefetch cache function is used.   |
|     |          |                  |     | [Clearing conditions]   |
|     |          |                  |     | Reset/standby   |
|     |          |                  |     | Writing a 0.  |
|     |          |                  |     | [Setting condition]   |
|     |          |                  |     | Writing a 1.  |
| 3   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0.   |
| 2   | PCE2     | 1                | R/W | Prefetch-Miss Cache Enable  |
|     |          |                  |     | Specifies usage or non-usage of the prefetch-miss cache.  |
|     |          |                  |     | 0: The prefetch-miss cache function is not used.  |
|     |          |                  |     | 1: The prefetch-miss cache function is used.  |
|     |          |                  |     | [Clearing conditions]   |
|     |          |                  |     | Reset/standby   |
|     |          |                  |     | Writing a 0.  |
|     |          |                  |     | [Setting condition]   |
|     |          |                  |     | Writing a 1.  |
|     |          |                  |     |   |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 1   | _        | 0                | R   | Reserved  |
|     |          |                  |     | This bit is always read as 0. The write value should always be 0. |
| 0   | PCE0     | 1                | R/W | Data Cache Enable   |
|     |          |                  |     | Specifies usage or non-usage of the data cache.                   |
|     |          |                  |     | 0: The data cache function is not used.                           |
|     |          |                  |     | 1: The data cache function is used.                               |
|     |          |                  |     | [Clearing conditions]   |
|     |          |                  |     | Reset/standby   |
|     |          |                  |     | Writing a 0.  |
|     |          |                  |     | [Setting condition]   |
|     |          |                  |     | Writing a 1.  |

## 27.4 Operation

### 27.4.1 Data Cache Lookup

When both ROM caching and the data cache are enabled (RCCR.RCE = 1 and RCCR2.PCE0 = 1) and data is read from the ROM area, the data cache is checked to see if it holds valid target data. The tag addresses of all four lines are simultaneously fed to four comparators, each of which compares bits 22 to 4 of one tag address with the access address. If the result of comparison is a match and the compared line is valid (the V bit = 1), the cache has been "hit" and LW0 to LW3 in the corresponding line of the data array are read out. Otherwise, the result is regarded as a cache miss.

Bits 3 and 2 of the access address then indicate the target longword for output to the CPU: 00 corresponds to LW0, 01 to LW1, 10 to LW2, and 11 to LW3.

Figure 27.2 shows the concept of looking up the data cache, with line 1 being hit.

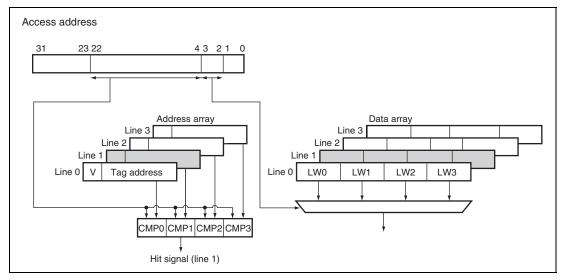


Figure 27.2 Concept of Looking up the Data Cache (with Line 1 being Hit)

If the data cache has not been hit (the cache was missed), data for the cache line is read out from the corresponding actual addresses in the ROM and bits 22 to 4 of the target address replace the tag address for the least recently used line (LRU method) of the address array. At the same time, the V bit is set and data from the target address is also output to the CPU.

### 27.4.2 Instruction Cache Lookup

In looking up the prefetch cache, whether the value of bit 4 of the access address is 0 or 1 determines whether the even- (0, 2, 4, or 6) or odd-numbered (1, 3, 5, or 7) lines should be checked, respectively. The tag addresses of the four selected lines are then simultaneously fed to four comparators, each of which compares bits 22 to 5 of one tag address with the access address. If the result of comparison is a match and the compared line is valid (the V bit = 1), the cache has been hit and LW0 to LW3 in the corresponding line of the data array are read out. Otherwise, the result is regarded as a cache miss.

Bits 3 and 2 of the access address then indicate the target longword for output to the CPU: 00 corresponds to LW0, 01 to LW1, 10 to LW2, and 11 to LW3.

Figure 27.3 shows the concept of looking up the prefetch cache, with line 2 being hit.

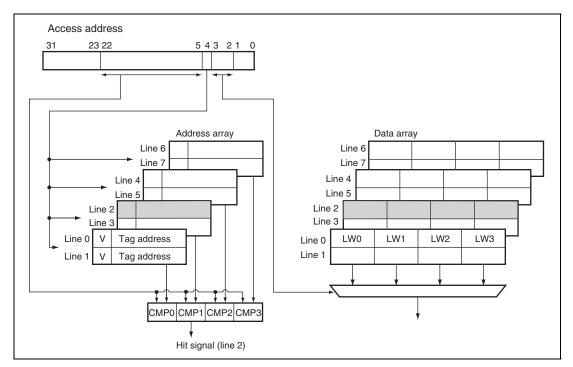


Figure 27.3 Concept of Looking up the Prefetch Cache (with Line 2 being Hit)

The prefetch-miss cache is looked up in the same way as the data cache. When a cache miss is produced by looking up both the prefetch and prefetch-miss cache, data for the cache line is read out from the corresponding actual addresses in the ROM and bits 22 to 4 of the target address

replace the tag address for the least recently used line (LRU method) of the address array. At the same time, the V bit is set and the instruction from the target address is also output to the CPU.

The prefetch cache is constantly updated if hardware prefetching is enabled. Whether bit 4 of the address containing the prefetched instruction is 0 or 1 determines whether the line to be updated is the least recently used of the even- or odd-numbered lines, respectively. Updating is done by replacing the tag address of the least recently used line with bits 22 to 5 of the read address and reading the data for the corresponding line of the data array from the ROM.

### 27.4.3 Hardware Prefetching

The ROM cache improves the hit rate of its component caches by hardware prefetching, including fetching in anticipation of the results of branch instructions. Prefetching is intended to prevent stalling in fetching by the processor, i.e. to ensure that the cache constantly supplies the processor with instructions to execute. There are two types of prefetching: consecutive prefetching and branch prefetching.

### (1) Consecutive Prefetching

Instructions that are subject to consecutive access are prefetched and placed in the prefetch cache.

### (2) Branch Prefetching

In branch prefetching, branch instructions are prefetched and decoded, their target addresses are predicted, and the instructions at those addresses are prefetched. Branch prefetching is subdivided into prefetching from the targets of conditional branches and prefetching from the targets of unconditional branches.

- Conditional-branch prefetching
   On encountering a conditional branch instruction (BF, BT, BF/S, or BT/S), prefetching from the predicted destination address is performed.
- · Unconditional-branch prefetching
  - BRA and BSR instructions

If unconditional-branch prefetching has been selected, prefetching always proceeds on encountering these unconditional branch instructions.

• JMP, JSR, and JSR/N instructions

If unconditional-branch prefetching has been selected, prefetching from the destinations of branch instructions of the above type is performed when the instruction has a register index and the index is fully predictable.

The following is an example of instructions that will lead to prefetching from the destination address of the JMP instruction when consecutive-instructive and unconditional-branch prefetching have been selected (JMP):

MOVI20 #imm20, Rn JMP @Rn

# 27.5 Usage Note

If a register of the ROMC is written to while the ROM is read by the DMAC or AUD-II, data may not be read correctly from the ROM. Do not write to the ROMC registers when the ROM is being accessed by the DMAC or AUD-II.

# Section 28 RAM

This LSI incorporates 256-Kbyte RAM, which is connected to F (CPU instruction Fetch), M (Memory access), and I (Internal) buses. This on-chip RAM can be accessed via any of these buses independently.

Figure 28.1 shows RAM block diagrams and figure 28.2 shows RAM and bus connections.

The on-chip RAM is allocated in addresses H'FFF80000 to H'FFFBFFFF (pages 0 to 15), as shown in table 28.1.

#### 28.1 **Features**

#### Access

The CPU/FPU, DMAC, A-DMAC, and AUD-II can access on-chip RAM in 8, 16, or 32 bits. Data in the on-chip RAM can be effectively used as program area or stack area data necessary for access at high speed.

For reading from the on-chip RAM, 1 or 2 cycles should be specified by a register according to the operating frequency (PLL multiplication ratio). For writing to the on-chip RAM, 2 or 3 cycles must be specified by a register according to the operating frequency (PLL multiplication ratio).

#### RAM data retention

The data in pages 1 and 0, a 32-Kbyte area, of RAM is retained during hardware standby.

#### ECC

The ECC error correction function can be enabled or disabled by register settings. The function is initially enabled. When the ECC is enabled, correction of one erroneous bit and detection of 2 erroneous bits are possible for 32 bits of data. When the ECC function is disabled, parity error detection is available. These ECC error detection and correction and parity errors are collectively called RAM errors. Flags that indicate occurrence of these RAM errors are provided.

#### Interrupts

Whether or not an interrupt is requested upon occurrence of a RAM error can be selected by register settings.

#### Ports

Each page in the on-chip RAM has two independent read and write ports. The read port is connected to I, F, and M buses and the write port is connected to I and M buses. The F and M buses are used for accesses from the CPU. The I bus is used for accesses from other than the CPU.

#### Priority

If the same page is accessed from multiple buses simultaneously, the access is performed according to the bus priority. The bus priority is as follows: I bus (highest), M bus (middle), F bus (lowest).

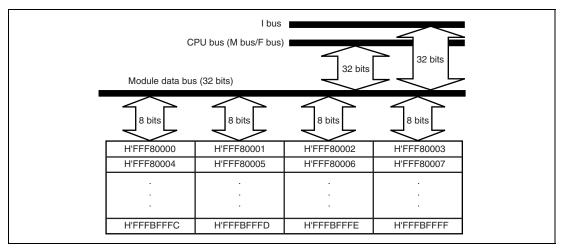


Figure 28.1 RAM Block Diagram

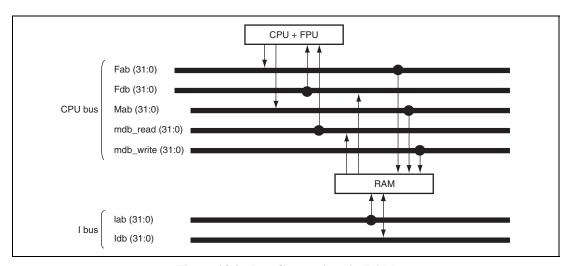


Figure 28.2 Bus Connections in RAM

# Table 28.1 On-chip RAM Address Space

| Page    | Address                  |
|---------|--------------------------|
| Page 0  | H'FFF80000 to H'FFF83FFF |
| Page 1  | H'FFF84000 to H'FFF87FFF |
| Page 2  | H'FFF88000 to H'FFF8BFFF |
| Page 3  | H'FFF8C000 to H'FFF8FFFF |
| Page 4  | H'FFF90000 to H'FFF93FFF |
| Page 5  | H'FFF94000 to H'FFF97FFF |
| Page 6  | H'FFF98000 to H'FFF9BFFF |
| Page 7  | H'FFF9C000 to H'FFF9FFFF |
| Page 8  | H'FFFA0000 to H'FFFA3FFF |
| Page 9  | H'FFFA4000 to H'FFFA7FFF |
| Page 10 | H'FFFA8000 to H'FFFABFFF |
| Page 11 | H'FFFAC000 to H'FFFAFFFF |
| Page 12 | H'FFFB0000 to H'FFFB3FFF |
| Page 13 | H'FFFB4000 to H'FFFB7FFF |
| Page 14 | H'FFFB8000 to H'FFFBBFFF |
| Page 15 | H'FFFBC000 to H'FFFBFFFF |

#### 28.2 **Register Descriptions**

The on-chip RAM has registers shown in table 28.2.

**Table 28.2 Register Configuration** 

| Register Name                        | Abbreviation | R/W | Initial Value | Address    | Access<br>Size |
|--------------------------------------|--------------|-----|---------------|------------|----------------|
| RAM enable control register 0        | RAMEN0       | R/W | H'00FF        | H'FFFF0800 | 8, (16)        |
| RAM write enable control register 0  | RAMWEN0      | R/W | H'00FF        | H'FFFF0802 | 8, (16)        |
| RAM ECC enable control register      | RAMECC       | R/W | H'0000        | H'FFFF0804 | 8, (16)        |
| RAM error status register            | RAMERR       | R/W | H'00          | H'FFFF0806 | 8              |
| RAM error interrupt control register | RAMINT       | R/W | H'00          | H'FFFF0810 | 8              |
| RAM access cycle set register        | RAMACYC      | R/W | H'0000        | H'FFFF0812 | 8, (16)        |
| RAM enable control register 1        | RAMEN1       | R/W | H'00FF        | H'FFFF0840 | 8, (16)        |
| RAM write enable control register 1  | RAMWEN1      | R/W | H'00FF        | H'FFFF0842 | 8, (16)        |

Apr 01, 2014

#### 28.2.1 RAM Enable Control Registers 0 and 1 (RAMEN0 and RAMEN1)

The RAM enable control registers (RAMEN0 and RAMEN1) are 16-bit readable/writable registers that enable or disable the access to the on-chip RAM. RAMEN0 and RAMEN1 are initialized to H'00FF by a reset or in the standby state. They can be written to in words, and can be read in bytes or words.

If the RAME15 to RAME0 bits corresponding to the access page are set to 1, accessing the onchip RAM becomes enabled; while if the RAME bits are cleared to 0, the on-chip RAM cannot be accessed. In the access disabled state, an undefined data is read if the page is read or if an instruction in the page is fetched, and a write to the page is ignored. The initial values of the RAME bits are 1.

To rewrite the RAME15 to RAME0 bits in RAMEN0, word size data with upper byte as H'96 and lower byte as write data must be written. Similarly, to rewrite the RAME15 to RAME0 bits in RAMEN1, word size data with upper byte as H'95 and lower byte as write data must be written.

When the upper byte (bits 15 to 8) of RAMEN0 or RAMEN1 is read, H'00 is always read.

Any instructions to access the on-chip RAM must not be placed immediately after any instructions to write to RAMEN0 or RAMEN1. Otherwise, correct access to the on-chip RAM cannot be guaranteed.

To rewrite the RAME15 to RAME0 bits in RAMEN0, execute an instruction to read from RAMEN0 together with five or more NOP instructions immediately after the instruction to write to RAMEN0. To rewrite the RAME15 to RAME0 bits in RAMEN1, execute instructions for RAMEN1 in the same way as for RAMEN0.

#### RAMEN0

| Bit:           | 15    | 14            | 13    | 12    | 11    | 10    | 9     | 8     | 7         | 6         | 5         | 4         | 3         | 2         | 1         | 0         |
|----------------|-------|---------------|-------|-------|-------|-------|-------|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
|                |       | RN0KEY[7:0]   |       |       |       |       |       |       | RAME<br>7 | RAME<br>6 | RAME<br>5 | RAME<br>4 | RAME<br>3 | RAME<br>2 | RAME<br>1 | RAME<br>0 |
| Initial value: | 0     | 0 0 0 0 0 0 0 |       |       |       |       |       |       | 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       |

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 28.2.7, Notes on Register Access.

|         |             | Initial |       |   |
|---------|-------------|---------|-------|---|
| Bit     | Bit Name    | Value   | R/W   | Descriptions  |
| 15 to 8 | RN0KEY[7:0] | H'00    | R/(W) | These bits enable or disable write to the RAME bit.   |
|         |             |         |       | H'96: Enable write to bits RAME7 to RAME0. The write data is not retained and these bits are always read as H'00. |
|         |             |         |       | Other than H'96: Disable write to bits RAME7 to RAME0.  |
| 7       | RAME7       | 1       | R/W   | RAM Enable 7  |
|         |             |         |       | Enables or disables access to page 7 in the on-chip RAM.  |
|         |             |         |       | 0: Disables access to page 7 in the on-chip RAM   |
|         |             |         |       | 1: Enables access to page 7 in the on-chip RAM  |
|         |             |         |       | [Clearing condition]  |
|         |             |         |       | <ul> <li>Writing 0 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul>                             |
|         |             |         |       | [Setting conditions]  |
|         |             |         |       | Reset or standby  |
|         |             |         |       | Writing 1 (H'96 is written to the upper byte simultaneously.)   |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 6   | RAME6    | 1                | R/W | RAM Enable 6  |
|     |          |                  |     | Enables or disables access to page 6 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables access to page 6 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables access to page 6 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 5   | RAME5    | 1                | R/W | RAM Enable 5  |
|     |          |                  |     | Enables or disables access to page 5 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables access to page 5 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables access to page 5 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'96 is written to the upper byte</li> </ul>                      |
|     |          |                  |     | simultaneously.)  |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'96 is written to the upper byte simultaneously.)                         |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 4   | RAME4    | 1                | R/W | RAM Enable 4  |
|     |          |                  |     | Enables or disables access to page 4 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables access to page 4 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables access to page 4 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 3   | RAME3    | 1                | R/W | RAM Enable 3  |
|     |          |                  |     | Enables or disables access to page 3 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables access to page 3 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables access to page 3 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'96 is written to the upper byte simultaneously.)                         |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 2   | RAME2    | 1                | R/W | RAM Enable 2  |
|     |          |                  |     | Enables or disables access to page 2 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables access to page 2 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables access to page 2 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 1   | RAME1    | 1                | R/W | RAM Enable 1  |
|     |          |                  |     | Enables or disables access to page 1 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables access to page 1 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables access to page 1 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'96 is written to the upper byte simultaneously.)                         |

|     |          | Initial |     |   |
|-----|----------|---------|-----|---|
| Bit | Bit Name | Value   | R/W | Descriptions  |
| 0   | RAME0    | 1       | R/W | RAM Enable 0  |
|     |          |         |     | Enables or disables access to page 0 in the on-chip RAM.                              |
|     |          |         |     | 0: Disables access to page 0 in the on-chip RAM                                       |
|     |          |         |     | 1: Enables access to page 0 in the on-chip RAM  |
|     |          |         |     | [Clearing condition]  |
|     |          |         |     | <ul> <li>Writing 0 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |         |     | [Setting conditions]  |
|     |          |         |     | Reset or standby  |
|     |          |         |     | <ul> <li>Writing 1 (H'96 is written to the upper byte<br/>simultaneously.)</li> </ul> |

#### RAMEN1

| Bit:           | 15    | 14          | 13    | 12    | 11    | 10    | 9     | 8     | 7          | 6          | 5          | 4          | 3          | 2          | 1         | 0         |
|----------------|-------|-------------|-------|-------|-------|-------|-------|-------|------------|------------|------------|------------|------------|------------|-----------|-----------|
|                |       | RN1KEY[7:0] |       |       |       |       |       |       | RAME<br>15 | RAME<br>14 | RAME<br>13 | RAME<br>12 | RAME<br>11 | RAME<br>10 | RAME<br>9 | RAME<br>8 |
| Initial value: | 0     | 0           | 0     | 0     | 0     | 0     | 0     | 0     | 1          | 1          | 1          | 1          | 1          | 1          | 1         | 1         |
| R/W:           | R/(W) | R/(W)       | R/(W) | R/(W) | R/(W) | R/(W) | R/(W) | R/(W) | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        | R/W       | R/W       |

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 28.2.7, Notes on Register Access.

| Bit     | Bit Name    | Initial<br>Value | R/W   | Descriptions   |
|---------|-------------|------------------|-------|--|
| 15 to 8 | RN1KEY[7:0] | H'00             | R/(W) | These bits enable or disable write to the RAME bit.  |
|         |             |                  |       | H'95: Enable write to bits RAME15 to RAME8. The write data is not retained and these bits are always read as H'00. |
|         |             |                  |       | Other than H'95: Disable write to bits RAME15 to RAME8.  |
| 7       | RAME15      | 1                | R/W   | RAM Enable 15  |
|         |             |                  |       | Enables or disables access to page 15 in the on-chip RAM.  |
|         |             |                  |       | 0: Disables access to page 15 in the on-chip RAM   |
|         |             |                  |       | 1: Enables access to page 15 in the on-chip RAM  |
|         |             |                  |       | [Clearing condition]   |
|         |             |                  |       | <ul> <li>Writing 0 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul>                              |
|         |             |                  |       | [Setting conditions]   |
|         |             |                  |       | Reset or standby   |
|         |             |                  |       | <ul> <li>Writing 1 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul>                              |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 6   | RAME14   | 1                | R/W | RAM Enable 14   |
|     |          |                  |     | Enables or disables access to page 14 in the on-chip RAM.                             |
|     |          |                  |     | 0: Disables access to page 14 in the on-chip RAM                                      |
|     |          |                  |     | 1: Enables access to page 14 in the on-chip RAM                                       |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 5   | RAME13   | 1                | R/W | RAM Enable 13   |
|     |          |                  |     | Enables or disables access to page 13 in the on-chip RAM.                             |
|     |          |                  |     | 0: Disables access to page 13 in the on-chip RAM                                      |
|     |          |                  |     | 1: Enables access to page 13 in the on-chip RAM                                       |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'95 is written to the upper byte simultaneously.)                         |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 4   | RAME12   | 1                | R/W | RAM Enable 12   |
|     |          |                  |     | Enables or disables access to page 12 in the on-chip RAM.                             |
|     |          |                  |     | 0: Disables access to page 12 in the on-chip RAM                                      |
|     |          |                  |     | 1: Enables access to page 12 in the on-chip RAM                                       |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 3   | RAME11   | 1                | R/W | RAM Enable 11   |
|     |          |                  |     | Enables or disables access to page 11 in the on-chip RAM.                             |
|     |          |                  |     | 0: Disables access to page 11 in the on-chip RAM                                      |
|     |          |                  |     | 1: Enables access to page 11 in the on-chip RAM                                       |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | • Writing 0 (H'95 is written to the upper byte  |
|     |          |                  |     | simultaneously.)  |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'95 is written to the upper byte simultaneously.)                         |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 2   | RAME10   | 1                | R/W | RAM Enable 10   |
|     |          |                  |     | Enables or disables access to page 10 in the on-chip RAM.                             |
|     |          |                  |     | 0: Disables access to page 10 in the on-chip RAM                                      |
|     |          |                  |     | 1: Enables access to page 10 in the on-chip RAM                                       |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 1   | RAME9    | 1                | R/W | RAM Enable 9  |
|     |          |                  |     | Enables or disables access to page 9 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables access to page 9 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables access to page 9 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'95 is written to the upper byte simultaneously.)                         |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 0   | RAME8    | 1                | R/W | RAM Enable 8  |
|     |          |                  |     | Enables or disables access to page 8 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables access to page 8 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables access to page 8 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'95 is written to the upper byte<br/>simultaneously.)</li> </ul> |

### 28.2.2 RAM Write Enable Control Registers 0 and 1 (RAMWEN0 and RAMWEN1)

The RAM write enable control registers (RAMWEN0 and RAMWEN1) are 16-bit readable/writable registers that enable or disable the access to the on-chip RAM. RAMWEN0 and RAMWEN1 are initialized to H'00FF by a reset or in the standby state. They can be written to in words, and can be read in bytes or words.

If the RAMWE15 to RAMWE0 bits corresponding to the page to be accessed are set to 1, writing to the on-chip RAM becomes enabled; while if the RAMWE bits are cleared to 0, the on-chip RAM cannot be written to. In the access disabled state, a write to the page is ignored. The initial values of the RAMWE bits are 1.

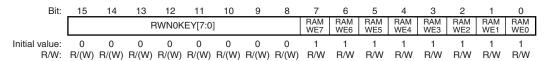
To rewrite the RAMWE15 to RAMWE0 bits in RAMWEN0, word size data with upper byte as H'69 and lower byte as write data must be written. Similarly, to rewrite the RAMWE15 to RAMWE0 bits in RAMWEN1, word size data with upper byte as H'68 and lower byte as write data must be written.

When the upper byte (bits 15 to 8) of RAMWEN0 or RAMWEN1 is read, the read value is always H'00.

Any instructions to access the on-chip RAM must not be placed immediately after any instructions to write to RAMWEN0 or RAMWEN1. Otherwise, correct access to the on-chip RAM cannot be guaranteed.

To rewrite the RAMWE15 to RAMWE0 bits in RAMWEN0, execute an instruction to read from RAMWEN0 together with five or more NOP instructions immediately after an instruction to write to RAMWEN0. To rewrite the RAMWE15 to RAMWE0 bits in RAMWEN1, execute instructions for RAMWEN1 in the same way as for RAMWEN0.

#### RAMWENO



Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 28.2.7, Notes on Register Access.

| Bit     | Bit Name | Initial<br>Value | R/W   | Descriptions  |
|---------|----------|------------------|-------|---|
| 15 to 8 | RWN0KEY  | H'00             | R/(W) | These bits enable or disable write to the RAMWE bit.  |
|         | [7:0]    |                  |       | H'69: Enable write to bits RAMWE7 to RAMWE0. The write data is not retained and these bits are always read as H'00. |
|         |          |                  |       | Other than H'69: Disable write to bits RAMWE7 to RAMWE0.  |
| 7       | RAMWE7   | 1                | R/W   | RAM Write Enable 7  |
|         |          |                  |       | Enables or disables write to page 7 in the on-chip RAM.   |
|         |          |                  |       | 0: Disables write to page 7 in the on-chip RAM  |
|         |          |                  |       | 1: Enables write to page 7 in the on-chip RAM.  |
|         |          |                  |       | [Clearing condition]  |
|         |          |                  |       | <ul> <li>Writing 0 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul>                               |
|         |          |                  |       | [Setting conditions]  |
|         |          |                  |       | Reset or standby  |
|         |          |                  |       | <ul> <li>Writing 1 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul>                               |
| 6       | RAMWE6   | 1                | R/W   | RAM Write Enable 6  |
|         |          |                  |       | Enables or disables write to page 6 in the on-chip RAM.   |
|         |          |                  |       | 0: Disables write to page 6 in the on-chip RAM  |
|         |          |                  |       | 1: Enables write to page 6 in the on-chip RAM   |
|         |          |                  |       | [Clearing condition]  |
|         |          |                  |       | Writing 0 (H'69 is written to the upper byte  |
|         |          |                  |       | simultaneously.)  |
|         |          |                  |       | [Setting conditions]  |
|         |          |                  |       | Reset or standby  |
|         |          |                  |       | <ul> <li>Writing 1 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul>                               |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 5   | RAMWE5   | 1                | R/W | RAM Write Enable 5  |
|     |          |                  |     | Enables or disables write to page 5 in the on-chip RAM.                               |
|     |          |                  |     | 0: Disables write to page 5 in the on-chip RAM  |
|     |          |                  |     | 1: Enables write to page 5 in the on-chip RAM   |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 4   | RAMWE4   | 1                | R/W | RAM Write Enable 4  |
|     |          |                  |     | Enables or disables write to page 4 in the on-chip RAM.                               |
|     |          |                  |     | 0: Disables write to page 4 in the on-chip RAM  |
|     |          |                  |     | 1: Enables write to page 4 in the on-chip RAM   |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'69 is written to the upper byte simultaneously.)                         |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 3   | RAMWE3   | 1                | R/W | RAM Write Enable 3  |
|     |          |                  |     | Enables or disables condition for write to page 3 in the on-chip RAM.                 |
|     |          |                  |     | 0: Disables write to page 3 in the on-chip RAM  |
|     |          |                  |     | 1: Enables write to page 3 in the on-chip RAM   |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 2   | RAMWE2   | 1                | R/W | RAM Write Enable 2  |
|     |          |                  |     | Enables or disables write to page 2 in the on-chip RAM.                               |
|     |          |                  |     | 0: Disables write to page 2 in the on-chip RAM  |
|     |          |                  |     | 1: Enables write to page 2 in the on-chip RAM   |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'69 is written to the upper byte simultaneously.)                         |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 1   | RAMWE1   | 1                | R/W | RAM Write Enable 1  |
|     |          |                  |     | Enables or disables write to page 1 in the on-chip RAM.                               |
|     |          |                  |     | 0: Disables write to page 1 in the on-chip RAM  |
|     |          |                  |     | 1: Enables write to page 1 in the on-chip RAM   |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 0   | RAMWE0   | 1                | R/W | RAM Write Enable 0  |
|     |          |                  |     | Enables or disables write to page 0 in the on-chip RAM.                               |
|     |          |                  |     | 0: Disables write to page 0 in the on-chip RAM  |
|     |          |                  |     | 1: Enables write to page 0 in the on-chip RAM   |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'69 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'69 is written to the upper byte simultaneously.)                         |

## RAMWEN1

| Bit:           | 15    | 14           | 13    | 12    | 11    | 10    | 9     | 8     | 7           | 6           | 5           | 4           | 3           | 2           | 1          | 0          |
|----------------|-------|--------------|-------|-------|-------|-------|-------|-------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|
|                |       | RWN1KEY[7:0] |       |       |       |       |       |       | RAM<br>WE15 | RAM<br>WE14 | RAM<br>WE13 | RAM<br>WE12 | RAM<br>WE11 | RAM<br>WE10 | RAM<br>WE9 | RAM<br>WE8 |
| Initial value: | 0     | 0            | 0     | 0     | 0     | 0     | 0     | 0     | 1           | 1           | 1           | 1           | 1           | 1           | 1          | 1          |
| R/W:           | R/(W) | R/(W)        | R/(W) | R/(W) | R/(W) | R/(W) | R/(W) | R/(W) | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W        | R/W        |

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 28.2.7, Notes on Register Access.

| Bit     | Bit Name | Initial<br>Value | R/W   | Descriptions   |
|---------|----------|------------------|-------|--|
| 15 to 8 | RWN1KEY  | H'00             | R/(W) | These bits enable or disable write to the RAMWE bit.   |
|         | [7:0]    |                  |       | H'68: Enable write to bits RAMWE15 to RAMWE8. The write data is not retained and these bits are always read as H'00. |
|         |          |                  |       | Other than H'68: Disable write to bits RAMWE15 to RAMWE8.  |
| 7       | RAMWE15  | 1                | R/W   | RAM Write Enable 15  |
|         |          |                  |       | Enables or disables write to page 15 in the on-chip RAM.   |
|         |          |                  |       | 0: Disables write to page 15 in the on-chip RAM  |
|         |          |                  |       | 1: Enables write to page 15 in the on-chip RAM.  |
|         |          |                  |       | [Clearing condition]   |
|         |          |                  |       | <ul> <li>Writing 0 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul>                                |
|         |          |                  |       | [Setting conditions]   |
|         |          |                  |       | Reset or standby   |
|         |          |                  |       | Writing 1 (H'68 is written to the upper byte simultaneously.)  |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 6   | RAMWE14  | 1                | R/W | RAM Write Enable 14   |
|     |          |                  |     | Enables or disables write to page 14 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables write to page 14 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables write to page 14 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 5   | RAMWE13  | 1                | R/W | RAM Write Enable 13   |
|     |          |                  |     | Enables or disables write to page 13 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables write to page 13 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables write to page 13 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'68 is written to the upper byte simultaneously.)                         |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 4   | RAMWE12  | 1                | R/W | RAM Write Enable 12   |
|     |          |                  |     | Enables or disables write to page 12 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables write to page 12 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables write to page 12 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 3   | RAMWE11  | 1                | R/W | RAM Write Enable 11   |
|     |          |                  |     | Enables or disables condition for write to page 11 in the on-chip RAM.                |
|     |          |                  |     | 0: Disables write to page 11 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables write to page 11 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'68 is written to the upper byte</li> </ul>                      |
|     |          |                  |     | simultaneously.)  |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 2   | RAMWE10  | 1                | R/W | RAM Write Enable 10   |
|     |          |                  |     | Enables or disables write to page 10 in the on-chip RAM.                              |
|     |          |                  |     | 0: Disables write to page 10 in the on-chip RAM                                       |
|     |          |                  |     | 1: Enables write to page 10 in the on-chip RAM  |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |
| 1   | RAMWE9   | 1                | R/W | RAM Write Enable 9  |
|     |          |                  |     | Enables or disables write to page 9 in the on-chip RAM.                               |
|     |          |                  |     | 0: Disables write to page 9 in the on-chip RAM  |
|     |          |                  |     | 1: Enables write to page 9 in the on-chip RAM   |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | Writing 1 (H'68 is written to the upper byte simultaneously.)                         |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 0   | RAMWE8   | 1                | R/W | RAM Write Enable 8  |
|     |          |                  |     | Enables or disables write to page 8 in the on-chip RAM.                               |
|     |          |                  |     | 0: Disables write to page 8 in the on-chip RAM  |
|     |          |                  |     | 1: Enables write to page 8 in the on-chip RAM   |
|     |          |                  |     | [Clearing condition]  |
|     |          |                  |     | <ul> <li>Writing 0 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | [Setting conditions]  |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 1 (H'68 is written to the upper byte<br/>simultaneously.)</li> </ul> |

## 28.2.3 RAM ECC Enable Control Register (RAMECC)

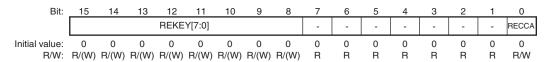
The RAMECC enable control register (RAMECC) enables or disables the ECC correction function. RAMECC is initialized by a reset or in the standby state.

RAMECC can be written to in words, and can be read in bytes or words. To rewrite to the RAMECC bits, word size data with upper byte as H'76 and lower byte as write data must be written.

When the upper byte (bits 15 to 8) of RAMECC is read, the read value is always H'00.

Do not place an instruction to access to the on-chip RAM immediately after an instruction to write to RAMECC; otherwise correct access to the on-chip RAM cannot be guaranteed.

When rewriting to the RECCA bit, execute an instruction to read from RAMECC together with five or more NOP instructions immediately after an instruction to write to RAMECC.



Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 28.2.7, Notes on Register Access.

|         |            | Initial |       |   |
|---------|------------|---------|-------|---|
| Bit     | Bit Name   | Value   | R/W   | Descriptions  |
| 15 to 8 | REKEY[7:0] | H'00    | R/(W) | These bits enable or disable write to the RECCA bit.  |
|         |            |         |       | H'76: Enable write to the RECCA bit. The write data is not retained and these bits are always read as H'00. |
|         |            |         |       | Other than H'76: Disable write to the RECCA bit.  |
| 7 to 1  | _          | All 0   | R     | Reserved  |
|         |            |         |       | These bits are always read as 0. The write value should always be 0.  |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 0   | RECCA    | 0                | R/W | Enables or disables the ECC correction.   |
|     |          |                  |     | 0: Enables ECC correction   |
|     |          |                  |     | [Clearing conditions]   |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 0 (H'76 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|     |          |                  |     | 1: Disables ECC correction  |
|     |          |                  |     | [Setting condition]   |
|     |          |                  |     | <ul> <li>Writing 1 (H'76 is written to the upper byte<br/>simultaneously.)</li> </ul> |

## 28.2.4 RAM Error Status Register (RAMERR)

The RAM error status register (RAMERR) monitors RAM error occurrence.

RAMERR is initialized by a reset or upon entering a standby state. RAMERR can be read or written to in bytes.

When the ECC error correction function is enabled, a 1-bit error correction in page 7 to page 0 sets the RCRTC0 bit and one in page 15 to page 8 sets the RCRTC1 bit during RAM read operation. 2-bit error detection in page 7 to page 0 sets the RDTCT0 bit and that in page 15 to page 8 sets the RDTCT1 bit. When the ECC error correction function is disabled, a parity error in page 7 to page 0 sets the RPARI0 bit and one in page 15 to page 8 sets the RPARI1 bit during an on-chip RAM read operation.

If the ECC error correction function is disabled via RAMECC after the RDTCT0 RDTCT1, RCRTC0, and RCRCT1 bits are set, these bits remain set. If the ECC error correction function is enabled via RAMECC after the RPARI0 and RPARI1 bits are set, these bits remain set.

Write H'00 to clear RAMERR. Immediately after the instruction for writing to RAMERR, execute an instruction to read from RAMERR and five or more NOP instructions. The status bits that have been set can be cleared by reading them while they are set and then writing 0 to them. Note that clearing a status bit by writing 0 is only possible after 1 has been read from it.

Bits 7 and 6 in RAMERR are always read as 0.

| Bit:           | 7 | 6 | 5          | 4          | 3          | 2          | 1          | 0          |
|----------------|---|---|------------|------------|------------|------------|------------|------------|
|                | - | - | RP<br>ARI1 | RP<br>ARI0 | RD<br>TCT1 | RC<br>RCT1 | RD<br>TCT0 | RC<br>RCT0 |
| Initial value: | 0 | 0 | 0          | 0          | 0          | 0          | 0          | 0          |
| R/W·           | R | R | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |

| Bit  | Bit Name | Initial<br>Value | R/W | Descriptions  |
|------|----------|------------------|-----|---|
| 7, 6 | _        | All 0            | R   | Reserved  |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 5    | RPARI1   | 0                | R/W | RAM parity error monitor bit 1  |
|      |          |                  |     | Monitors whether or not a parity error occurs in page 15 to page 8 when the ECC error correction is disabled. |
|      |          |                  |     | 0: No parity error has occurred.  |
|      |          |                  |     | 1: A parity error has occurred.   |
|      |          |                  |     | [Clearing conditions]   |
|      |          |                  |     | Reset or standby  |
|      |          |                  |     | <ul> <li>Writing 0 after reading 1</li> </ul>   |
|      |          |                  |     | [Setting condition]   |
|      |          |                  |     | A parity error has occurred   |
| 4    | RPARI0   | 0                | R/W | RAM parity error monitor bit 0  |
|      |          |                  |     | Monitors whether or not a parity error occurs in page 7 to page 0 when the ECC error correction is disabled.  |
|      |          |                  |     | 0: No parity error has occurred.  |
|      |          |                  |     | 1: A parity error has occurred.   |
|      |          |                  |     | [Clearing conditions]   |
|      |          |                  |     | Reset or standby  |
|      |          |                  |     | <ul> <li>Writing 0 after reading 1</li> </ul>   |
|      |          |                  |     | [Setting condition]   |
|      |          |                  |     | <ul> <li>A parity error has occurred</li> </ul>   |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions   |
|-----|----------|------------------|-----|--|
| 3   | RDTCT1   | 0                | R   | RAM 2-bit error detection monitor bit 1  |
|     |          |                  |     | Monitors whether or not 2-bit error detection occurs in page 15 to page 8 when the ECC error correction is enabled.    |
|     |          |                  |     | 0: No 2-bit error detection has occurred.  |
|     |          |                  |     | 1: 2-bit error detection has occurred.   |
|     |          |                  |     | [Clearing conditions]  |
|     |          |                  |     | Reset or standby   |
|     |          |                  |     | <ul> <li>Writing 0 after reading 1</li> </ul>  |
|     |          |                  |     | [Setting condition]  |
|     |          |                  |     | <ul> <li>2-bit error detection has occurred</li> </ul>   |
| 2   | RCRCT1   | 0                | R/W | RAM 1-bit error detection monitor bit 1  |
|     |          |                  |     | Monitors whether or not a 1-bit error correction occurs in page 15 to page 8 when the ECC error correction is enabled. |
|     |          |                  |     | 0: No 1-bit error correction has occurred.   |
|     |          |                  |     | 1: A 1-bit error correction has occurred.  |
|     |          |                  |     | [Clearing conditions]  |
|     |          |                  |     | Reset or standby   |
|     |          |                  |     | <ul> <li>Writing 0 after reading 1</li> </ul>  |
|     |          |                  |     | [Setting condition]  |
| -   |          |                  |     | A 1-bit error correction has occurred  |
| 1   | RDTCT0   | 0                | R/W | RAM 2-bit error detection monitor bit 0  |
|     |          |                  |     | Monitors whether or not 2-bit error detection occurs in page 7 to page 0 when the ECC error correction is enabled.     |
|     |          |                  |     | 0: No 2-bit error detection has occurred.  |
|     |          |                  |     | 1: 2-bit error detection has occurred.   |
|     |          |                  |     | [Clearing conditions]  |
|     |          |                  |     | Reset or standby   |
|     |          |                  |     | <ul> <li>Writing 0 after reading 1</li> </ul>  |
|     |          |                  |     | [Setting condition]  |
|     |          |                  |     | 2-bit error detection has occurred   |

| Bit | Bit Name | Initial<br>Value | R/W | Descriptions  |
|-----|----------|------------------|-----|---|
| 0   | RCRCT0   | 0                | R/W | RAM 1-bit error correction monitor bit 0  |
|     |          |                  |     | Monitors whether or not a 1-bit error correction occurs in page 7 to page 0 when the ECC error correction is enabled. |
|     |          |                  |     | 0: No 1-bit error correction has occurred.  |
|     |          |                  |     | 1: A 1-bit error correction has occurred.   |
|     |          |                  |     | [Clearing conditions]   |
|     |          |                  |     | Reset or standby  |
|     |          |                  |     | <ul> <li>Writing 0 after reading 1</li> </ul>   |
|     |          |                  |     | [Setting condition]   |
|     |          |                  |     | <ul> <li>A 1-bit error correction has occurred</li> </ul>   |

## 28.2.5 RAM Error Interrupt Control Register (RAMINT)

The RAM error interrupt control register (RAMINT) enables or disables the RAM error interrupt.

If the RECIE0 and RECIE1 bits of RAMINT are set enabled while the ECC error correction function is set enabled via RAMECC, an interrupt is generated upon the correction of a 1-bit error or detection of a 2-bit error. Also, if the REDIE0 and REDIE1 bits of RAMINT are set enabled, an interrupt is generated upon detection of a 2-bit error. Table 28.3 shows the conditions required for interrupt generation when the ECC error correction is set enabled. If the RPEIE0 and RPEIE1 bits of RAMINT are set enabled while the ECC error correction function is set enabled via RAMECC, an interrupt is generated upon occurrence of a parity error.

RECIE0, REDIE0, and RPEIE0 correspond to the RAM area of page 7 to page 0. Also, RECIE1, REDIE1, and RPEIE1 correspond to that of page 15 to page 0.

RAMINT is initialized by a reset or upon entering the standby state. RAMINT can only be read or written to in bytes. When rewriting RAMINT, execute an instruction to read from RAMINT together five or more NOP instructions immediately after an instruction to write to RAMINT.

| Bit:           | 7 | 6 | 5          | 4          | 3          | 2          | 1          | 0          |  |
|----------------|---|---|------------|------------|------------|------------|------------|------------|--|
|                | - | - | RP<br>EIE1 | RP<br>EIE0 | RE<br>DIE1 | RE<br>CIE1 | RE<br>DIE0 | RE<br>CIE0 |  |
| Initial value: | 0 | 0 | 0          | 0          | 0          | 0          | 0          | 0          |  |
| R/W:           | R | R | R/W        | R/W        | R/W        | R/W        | R/W        | R/W        |  |

| Bit  | Bit Name | Initial<br>Value | R/W | Descriptions  |
|------|----------|------------------|-----|---|
| 7, 6 | _        | All 0            | R   | Reserved  |
|      |          |                  |     | These bits are always read as 0. The write value should always be 0.  |
| 5    | RPEIE1   | 0                | R/W | RAM parity error interrupt bit 1  |
|      |          |                  |     | Enables/disables an interrupt upon occurrence of a parity error in page 15 to page 8 when the ECC error correction is disabled. |
|      |          |                  |     | 0: Disables an interrupt upon occurrence of a parity error.   |
|      |          |                  |     | Enables an interrupt upon occurrence of a parity error.   |

|     |          | Initial |     |  |
|-----|----------|---------|-----|--|
| Bit | Bit Name | Value   | R/W | Descriptions   |
| 4   | RPEIE0   | 0       | R/W | RAM parity error interrupt bit 0   |
|     |          |         |     | Enables/disables an interrupt upon occurrence of a parity error in page 7 to page 0 when the ECC error correction is disabled.           |
|     |          |         |     | <ol><li>Disables an interrupt upon occurrence of a parity<br/>error.</li></ol>   |
|     |          |         |     | <ol> <li>Enables an interrupt upon occurrence of a parity<br/>error.</li> </ol>  |
| 3   | REDIE1   | 0       | R/W | RAM 2-bit error detection interrupt bit 1  |
|     |          |         |     | Enables/disables an interrupt upon detection of a 2-bit error in page 15 to page 8 when the ECC error correction is enabled.             |
|     |          |         |     | 0: Disables an interrupt upon detection of a 2-bit error.  |
|     |          |         |     | 1: Enables an interrupt upon detection of a 2-bit error.   |
| 2   | RECIE1   | 0       | R/W | RAM 1-bit error correction interrupt bit 1   |
|     |          |         |     | Enables/disables an interrupt upon occurrence of a 1-bit error correction in page 15 to page 8 when the ECC error correction is enabled. |
|     |          |         |     | 0: Disables an interrupt upon occurrence of a 1-bit error correction.  |
|     |          |         |     | 1: Enables an interrupt upon occurrence of a 1-bit error correction.   |

| D:4 | Dit Name | Initial<br>Value | D/M | Descriptions  |
|-----|----------|------------------|-----|---|
| Bit | Bit Name | value            | R/W | Descriptions  |
| 1   | REDIE0   | 0                | R/W | RAM 2-bit error detection interrupt bit 0   |
|     |          |                  |     | Enables/disables an interrupt upon detection of a 2-bit error in page 7 to page 0 when the ECC error correction is enabled.             |
|     |          |                  |     | 0: Disables an interrupt upon detection of a 2-bit error.   |
|     |          |                  |     | 1: Enables an interrupt upon detection of a 2-bit error.  |
| 0   | RECIE0   | 0                | R/W | RAM 1-bit error correction interrupt bit 0  |
|     |          |                  |     | Enables/disables an interrupt upon occurrence of a 1-bit error correction in page 7 to page 0 when the ECC error correction is enabled. |
|     |          |                  |     | 0: Disables an interrupt upon occurrence of a 1-bit error correction.   |
|     |          |                  |     | 1: Enables an interrupt upon occurrence of a 1-bit error correction.  |

Table 28.3 Conditions for Interrupt Occurrence When ECC Error Correction is Enabled

| REDIE Bit | RECIE Bit | RAM Errors to Trigger an Interrupt                                   |
|-----------|-----------|--|
| 0         | 0         | None   |
| 0         | 1         | Upon occurrence of a 1-bit error correction or 2-bit error detection |
| 1         | 0         | Upon occurrence of 2-bit error correction                            |
| 1         | 1         | Upon occurrence of a 1-bit error correction or 2-bit error detection |

## 28.2.6 RAM Access Cycle Set Register (RAMACYC)

The RAM access cycle set register (RAMACYC) sets the RAM read or write cycle. The cycle setting range depends on the PLL clock multiplication ratio and ECC enable/disable. For details, refer to the recommended setting range shown in table 28.4.

The WRCYC and RDCYC bits are initialized by a reset or in the standby state.

To write the RAMACYC bits, word size data with upper byte as H'78 and lower byte as write data must be written. RAMACYC can be written to in words, and can be read in bytes or words.

When the upper byte (bits 15 to 8) of RAMACYC is read, the read value is always H'00.

Do not write to RAMACYC while RAM is being accessed. Be sure to set all the RAME bits in the RAM enable control register to 0 to disable access to the RAM before rewriting RAMACYC. Also, do not place an instruction to access to the on-chip RAM immediately after an instruction to write to RAMACYC; otherwise normal access is not guaranteed.

When rewriting the WRCYC1, WRCYC0 and RDCYC bits, execute an instruction to read from RAMACYC together with five or more NOP instructions immediately after an instruction to write to RAMACYC.

| Bit:           | 15    | 14    | 13    | 12    | 11     | 10    | 9     | 8     | 7 | 6 | 5   | 4       | 3 | 2 | 1 | 0     |
|----------------|-------|-------|-------|-------|--------|-------|-------|-------|---|---|-----|---------|---|---|---|-------|
|                |       |       |       | RAKE  | Y[7:0] |       |       |       | - | - | WRC | /C[1:0] | - | 1 | - | RDCYC |
| Initial value: | 0     | 0     | 0     | 0     | 0      | 0     | 0     | 0     | 0 | 0 | 0   | 0       | 0 | 0 | 0 | 0     |
| R/W:           | R/(W) | R/(W) | R/(W) | R/(W) | R/(W)  | R/(W) | R/(W) | R/(W) | R | R | R/W | R/W     | R | R | R | R/W   |

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 28.2.7, Notes on Register Access.

| Bit     | Bit Name   | Initial<br>Value | R/W   | Descriptions   |
|---------|------------|------------------|-------|--|
| 15 to 8 | RAKEY[7:0] | H'00             | R/(W) | These bits enable or disable write to bits WRCYC[1], WRCYC[0], and RDCYC.  |
|         |            |                  |       | H'78: Enable write to bits WRCYC[1], WRCYC[0], and RDCYC. The write data is not retained and these bits are always read as H'00. |
|         |            |                  |       | Other than H'78: Disable write to bits WRCYC[1], WRCYC[0], and RDCYC.  |
| 7, 6    | _          | All 0            | R     | Reserved   |
|         |            |                  |       | These bits are always read as 0. The write value should always be 0.   |

| Bit    | Bit Name   | Initial<br>Value | R/W | Descriptions   |
|--------|------------|------------------|-----|--|
| 5, 4   | WRCYC[1:0] | 00               | R/W | These bits set the RAM write cycle.  |
|        |            |                  |     | 00: Set write cycle to 4 cycles  |
|        |            |                  |     | 01: Set write cycle to 3 cycles  |
|        |            |                  |     | 10: Set write cycle to 2 cycles  |
|        |            |                  |     | 11: Setting prohibited   |
|        |            |                  |     | [Clearing conditions]  |
|        |            |                  |     | Reset or standby   |
|        |            |                  |     | <ul> <li>Writing B'00 (H'78 is written to the upper byte<br/>simultaneously.)</li> </ul> |
|        |            |                  |     | [Setting condition]  |
|        |            |                  |     | H'78 is written to the upper byte simultaneously.  |
|        |            |                  |     | Note: Do not set for 2 cycles unless $\times 4$ multiplication has been set for the PLL. |
| 3 to 1 | _          | All 0            | R   | Reserved   |
|        |            |                  |     | These bits are always read as 0. The write value should always be 0.                     |
| 0      | RDCYC      | 0                | R/W | Sets the RAM read cycle.   |
|        |            |                  |     | 0: Set read cycle to 2 cycles  |
|        |            |                  |     | [Clearing conditions]  |
|        |            |                  |     | Reset or standby   |
|        |            |                  |     | <ul> <li>Writing 0 (H'78 is written to the upper byte<br/>simultaneously.)</li> </ul>    |
|        |            |                  |     | 1: Set read cycle to 1 cycle   |
|        |            |                  |     | [Setting condition]  |
|        |            |                  |     | <ul> <li>Writing 1 (H'78 is written to the upper byte<br/>simultaneously.)</li> </ul>    |
|        |            |                  |     | Note: Do not set for 1 cycle unless ×4 multiplication has been set for the PLL.          |

Accesses through the I bus are ruled by the peripheral clock  $(P\phi)$ ; this is not the case with M or F buses, which are ruled by the internal clock  $(\phi)$ . Therefore the number of cycles required for an I-bus access is determined by combination of the set PLL and peripheral clock multiplication ratios as well as the set value in this register.

It takes one cycle to receive a command or address during I bus read access on condition that the PLL multiplication ratio is  $\times 4$  or  $\times 6$  or on condition that the PLL multiplication ratio is  $\times 8$  or  $\times 10$  and the peripheral clock multiplication ratio is  $\times 2$ . It also takes one cycle during I bus write access on condition that the PLL multiplication ratio is  $\times 4$  or  $\times 6$  and the peripheral clock multiplication ratio is  $\times 2$ . This is why the entire process, receiving a command and address, needs two cycles on such conditions.

Table 28.4 Recommended Settings for Read and Write Cycles

|                                     | Recomme          | ended Settings | -                              | I Bus A | Access (Pφ)                  |                 |                  |
|-------------------------------------|------------------|----------------|--------------------------------|---------|------------------------------|-----------------|------------------|
| PLL<br>Multipli-<br>cation<br>Ratio | RDCYC WRCYC[1:0] |                | Read Write<br> [Cycle] [Cycle] |         | Clock Multipli- cation Ratio | Read<br>[Cycle] | Write<br>[Cycle] |
| ×4                                  | B'1              | B'10           | 1                              | 2       | ×1                           | 2               | 1                |
|                                     |                  |                |                                |         | ×2                           | 2               | 2                |
| ×6                                  | B'0              | B'01           | 2                              | 3       | ×1                           | 2               | 1                |
|                                     |                  |                |                                |         | ×2                           | 2               | 2                |
| ×8                                  |                  |                | 2                              | 3       | ×1                           | 1               | 1                |
|                                     |                  |                |                                |         | ×2                           | 2               | 1                |
| ×10                                 | _                |                | 2                              | 3       | ×1                           | 1               | 1                |
|                                     |                  |                |                                |         | ×2                           | 2               | 1                |

## 28.2.7 Notes on Register Access

The way of writing data to the RAM enable control registers 0 and 1 (RAMEN0 and RAMEN1), RAM write enable control registers 0 and 1 (RAMWEN), RAM ECC enable control register (RAMECC), and RAM access cycle set register (RAMACYC) is different from that to other general registers. This is because these registers are not to be easily rewritten.

To write these registers, use the following ways. In addition, note that RAMENO, RAMEN1, RAMWENO, RAMWEN1, RAMECC, and RAMACYC must be written in words. These registers cannot be written in byte or longword instructions. As shown in figure 28.3, key data should be written in the upper byte.

- To write data to RAMENO, transfer data with upper byte as H'96 and lower byte as write data.
- To write data to RAMEN1, transfer data with upper byte as H'95 and lower byte as write data.
- To write data to RAMWEN0, transfer data with upper byte as H'69 and lower byte as write data.

To write data to RAMWEN1, transfer data with upper byte as H'68 and lower byte as write data.

- To write data to RAMECC, transfer data with upper byte as H'76 and lower byte as write data.
- To write data to RAMACYC, transfer data with upper byte as H'78 and lower byte as write data.

When the upper bytes (bits 15 to 8) of RAMEO, RAME1, RAMWENO, RAMWEN1, RAMECC, and RAMACYC are read, the read value is always H'00.

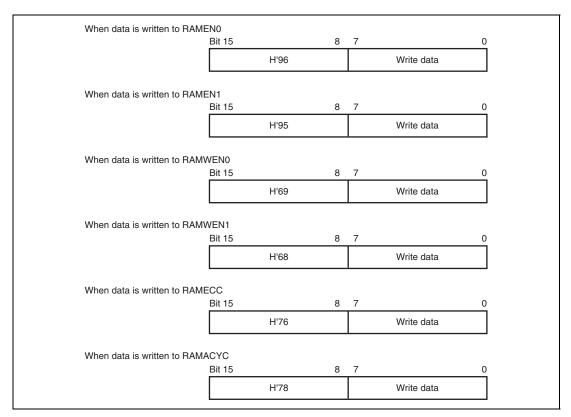


Figure 28.3 Writing Data to RAMEN0, RAMEN1, RAMWEN0, RAMWEN1, RAMECC, and RAMACYC

## 28.3 On-Chip RAM Operations

Access to the on-chip RAM is controlled by the RAM enable control registers 0 and 1 (RAMEN0 and RAMEN1) and RAM write enable control registers 0 and 1 (RAMWEN0 and RAMWEN1).

Accessing each area of the on-chip RAM is enabled or disabled by the RAME15 to RAME0 bits in the RAM enable control registers 0 and 1 (RAMEN0 and RAMEN1). When the RAME15 to RAME0 bits of RAMEN0 and RAMEN1 are cleared to 0, the on-chip RAM cannot be accessed. In this case, values read from the on-chip RAM are undefined and the on-chip RAM cannot be modified.

Writing to each area of the on-chip RAM is enabled or disabled by the RAMWE15 to RAMWE0 bits in the RAM write enable control registers 0 and 1 (RAMWEN0 and RAMWEN1).

The ECC error correction function can be set enabled or disabled by register settings. The function is initially set enabled.

When the ECC error correction function is set enabled, 1-bit error correction and 2-bit error detection can be performed. Upon correction of a 1-bit error and/or detection of a 2-bit error, flags (in the RAM error status register) are set to indicate their occurrence. When the ECC error correction is set disabled, a flag (in the RAM error status register) is set upon occurrence of a parity error. Whenever the RAM error status register is set, a RAM error (RAME) interrupt can be generated. The interrupt generation can be set enabled or disabled via the RAM error interrupt control register (RAMINT).

## 28.4 RAM Data Retention

### 28.4.1 Data Retention at Reset

If a low level signal is input on the RES pin from an external device while this LSI is in operation, this LSI enters the power-on reset state. In this case, if the on-chip RAM is accessed, data in the RAM address being accessed may be destroyed because the bus cycle cannot be completed normally.

Since it is difficult to input reset signals from the external devices while avoiding accesses to the on-chip RAM. Accordingly, to retain all data items in the on-chip RAM at reset, invalidate the RAM by the RAM enable control register (RAMEN).

However, data in the on-chip RAM is not retained when the operation enters boot mode or user boot mode from the power-on reset state since the RAM is cannot be disabled as occupied by programs embedded in the LSI.

## 28.4.2 Data Retention at Hardware Standby

This LSI turns off the internal power supply except for pages 0 and 1 in the on-chip RAM if the hardware standby state is entered. Accordingly, data in the on-chip RAM other than that in pages 0 and 1 cannot be retained. To retain data in pages 0 and 1 during hardware standby state, hardware standby state must be entered according to the sequence shown in figure 28.4.

During hardware standby state, data in pages 0 and 1 can be retained while the specified voltage  $(V_{RAM})$  is supplied to the Vcc power supply.

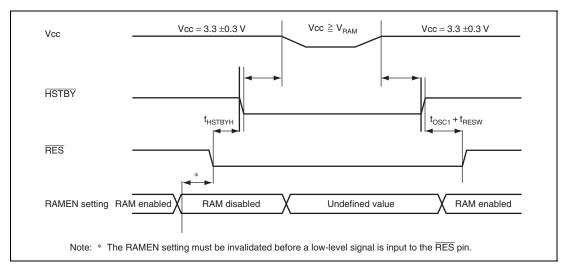


Figure 28.4 RAM Data Retention at Hardware Standby

## 28.5 Notes on Usage

## 28.5.1 Page Conflict

If the same page is accessed by the different buses simultaneously, a page conflict occurs. Each of those accesses is handled in such priority scheme as: I bus (highest), M bus (middle), F bus (lowest).

In this case, each access is completed normally but this conflict degrades the memory access efficiency. To avoid this conflict, it is recommended to take preventative measures by software. For example, accessing different memory or different pages using different buses can avoid page conflict.

## 28.5.2 State After Turning on Power

After turning on the power, all data items in the on-chip RAM including ECC correction data and parity are undefined. Accordingly, correspondence among RAM data, error correction data, and parity may not be correct.

To enable correct (initialize) correspondence among RAM data, error correction data, and parity after power-on, write data to all on-chip RAM areas. If RAM is read without initialization, a RAM error may occur. Note that no RAM error occurs in RAM writes.

## 28.5.3 Write Operation When Writing RAM is Disabled

When a write operation is performed to a page with RAM writing disabled in initialized RAM and the address of the page has encountered an ECC error, an ECC error flag is set due to wrongly detected ECC error. Also interrupts are generated if interrupts are enabled. In that case, however, the data are not destroyed since the write operation is not performed.

## Section 29 Power-Down Modes

To reduce device power consumption, this device incorporates two types of power down modes: hardware standby mode and sleep mode, and a module standby function for halting the operation of certain modules. A proper mode or function should be selected according to the application.

## 29.1 Features

#### 29.1.1 Power-Down Modes

This LSI has the following power-down modes and function:

- Hardware standby mode
  - This LSI enters hardware standby mode by inputting specific levels to the RES and HSTBY pins. In this mode, all the functions of the LSI halt and power supply to most part of the internal LSI is halted. The LSI returns from hardware standby mode by a power-on reset.
- Sleep mode
  - The LSI enters sleep mode by the CPU instruction. In this mode, the on-chip peripheral modules other than the CPU operate. The LSI returns from sleep mode by a power-on reset, an interrupt, or a DMA address error.
- Module standby function
  - The operation of on-chip peripheral modules that can enter the module standby state (FPU, UBC, DMAC, AUD-II, and JTAG Interface) is halted by stopping clock supply. The clock supply to the target module can be controlled by the corresponding bit of STBCR.

Table 29.1 shows states of the CPU and peripheral modules in each mode. Table 29.2 shows the transition conditions for entering each mode from the program execution state, as well as the procedures for revoking each mode.

**Table 29.1 States of Power-Down Modes** 

|                          |                                   |                                   | State*                            |  |                                   |
|--------------------------|-----------------------------------|-----------------------------------|-----------------------------------|--|-----------------------------------|
| Power-Down<br>Mode       | CPG                               | CPU                               | CPU Register                      | On-Chip RAM  | On-Chip<br>Peripheral<br>Modules  |
| Hardware<br>standby mode | Halts<br>(Power supply<br>halted) | Halts<br>(Power supply<br>halted) | Halts<br>(Power supply<br>halted) | A part of RAM area (32 Kbytes) is retained. (Power supply halted outside the retained area.) | Halts<br>(Power supply<br>halted) |
| Sleep mode               | Runs                              | Halts                             | Held                              | Runs   | Runs                              |
| Module standby           | Runs                              | Runs                              | Runs                              | Runs   | Specified module halts            |

Note: \* The pin state is retained or set to high impedance. For details, see appendix A, Pin States.

Table 29.2 Transition Conditions for Entering Power-Down Mode and Revoking Procedures

| Power-Down Mode         | Transition Conditions  | Revoking Procedure  |  |  |  |
|-------------------------|--|---|--|--|--|
| Hardware standby mode   | <ul> <li>Driving the HSTBY pin low<br/>while the RES pin is low</li> </ul> | Power-on reset after inputting<br>a high level signal on the<br>HSTBY pin |  |  |  |
| Sleep mode              | Executing SLEEP instruction  | Power-on reset  |  |  |  |
|                         |  | <ul> <li>Interrupt</li> </ul>   |  |  |  |
|                         |  | <ul> <li>DMA address error</li> </ul>                                     |  |  |  |
| Module standby function | Setting the MSTP bits of   | Clearing the MSTP bit to 0  |  |  |  |
|                         | STBCR to 1   | <ul> <li>Power-on reset</li> </ul>  |  |  |  |

function

## 29.2 Input/Output Pins

Table 29.3 shows the pin configuration related to power-down modes.

**Table 29.3 Pin Configuration** 

| Pin Name         | Symbol | I/O   | Function  |
|------------------|--------|-------|---|
| Power-on reset   | RES    | Input | Inputting a low level signal on this pin causes a transition to power-on reset processing.                      |
| Hardware standby | HSTBY  | Input | Inputting a low level signal on this pin while the RES pin is low causes a transition to hardware standby mode. |

## 29.3 Register Descriptions

The following register is used in power-down modes.

**Table 29.4 Register Configuration** 

| Register Name            | Abbreviation | R/W | Initial Value | Address    | Access<br>Size |
|--------------------------|--------------|-----|---------------|------------|----------------|
| Standby control register | STBCR        | R/W | H'0000        | H'FFFE0400 | 8, 16          |

## 29.3.1 Standby Control Register (STBCR)

STBCR is a 16-bit readable/writable register that controls the operation of each module in power-down modes. This register is initialized to H'0000 by a power-on reset or in hardware standby mode.

STBCR must be written to in word units. When rewriting the values of the MSTP4 to MSTP0 bits, write H'3C to the STBCRKEY bits simultaneously. If data other than H'3C is written to the STBCRKEY bits or data is written in byte units, the operation is ignored.

Both word and byte accesses are available to read STBCR. However, the STBCRKEY bits are always read as H'00 because the values written to the STBCRKEY bits are not retained.

Note: To prevent erroneous rewriting, the way of writing data to this register is different from that for other general registers. For details, refer to section 29.3.2, Note on Accessing STBCR.

| Bit:           | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    | 7 | 6 | 5 | 4     | 3     | 2     | 1     | 0     |
|----------------|------|------|------|------|------|------|------|------|---|---|---|-------|-------|-------|-------|-------|
|                |      |      |      | STBC | RKEY |      |      |      | - | - | - | MSTP4 | MSTP3 | MSTP2 | MSTP1 | MSTP0 |
| Initial value: | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0 | 0 | 0 | 0     | 0     | 0     | 0     | 0     |
| R/W:           | R/W* | R | R | R | R/W   | R/W   | R/W   | R/W   | R/W   |

Note: \* The write value is not retained.

| Bit     | Bit Name | Initial<br>Value | R/W | Description  |
|---------|----------|------------------|-----|--|
| 15 to 8 | STBCRKEY | H'00             | R/W | STBCR Write Key Code   |
|         |          |                  |     | These bits enable or disable modifying bits MSTP4 to MSTP0. These bits are always read as H'00 because the values written to these bits are not retained.  |
|         |          |                  |     | H'3C: Enable modifying bits MSTP4 to MSTP0   |
|         |          |                  |     | Other than H'3C: Disable modifying bits MSTP4 to MSTP0   |
| 7 to 5  | _        | All 0            | R   | Reserved   |
|         |          |                  |     | These bits are always read as 0. The write value should always be 0.   |
| 4       | MSTP4    | 0                | R/W | Module Stop 4  |
|         |          |                  |     | Setting this bit to 1 puts the JTAG TAP controller into a reset state. Clearing this bit to 0 releases the TAP controller from its reset state and brings back into operation in accordance with input on the JTAG pins.                               |
|         |          |                  |     | 0: The JTAG TAP module is in operation.  |
|         |          |                  |     | 1: The JTAG TAP module is in a reset state.  |
| 3       | MSTP3    | 0                | R/W | Module Stop 3  |
|         |          |                  |     | When this bit is set to 1, the clock supply to the DMAC is halted. When this bit is cleared to 0, the clock supply to the DMAC is restarted. Note that the registers in the DMAC are not initialized even when the clock supply to the DMAC is halted. |
|         |          |                  |     | 0: DMAC runs   |
|         |          |                  |     | 1: Clock supply to DMAC is halted  |

| Bit | Bit Name | Initial<br>Value | R/W | Description   |
|-----|----------|------------------|-----|---|
| 2   | MSTP2    | 0                | R/W | Module Stop 2   |
| _   |          | v                |     | When the MSTP2 bit is set to 1, the clock supply to the AUD-II is halted. When this bit is cleared to 0, the clock supply to the AUD-II is restarted. The internal state (including registers) of the AUD-II is initialized when the clock supply to the AUD-II is halted.  |
|     |          |                  |     | However, if the trace function is enabled (the EN bit of AUCSR is set to 1) while the AUD-II operates in trace mode, the AUD-II does not enter the module standby state. Even if the trace function is disabled, the AUD-II does not enter the module standby state as long as there remains acquired trace data in the internal FIFO. The AUD-II enters the module standby state after all remaining data has been output from the FIFO. |
|     |          |                  |     | 0: AUD-II runs  |
|     |          |                  |     | 1: Clock supply to AUD-II is halted   |
| 1   | MSTP1    | 0                | R/W | Module Stop 1   |
|     |          |                  |     | When this bit is set to 1, the clock supply to the FPU is halted. Once this bit is set to 1, it is impossible to write 0 to this bit; that is to say, once the clock supply to the FPU is halted by setting this bit to 1, the clock supply to the FPU cannot be restarted by clearing this bit to 0. To restart the clock supply to the FPU, reset the LSI by a power-on reset.  |
|     |          |                  |     | 0: FPU runs   |
|     |          |                  |     | 1: Clock supply to FPU is halted  |
| 0   | MSTP0    | 0                | R/W | Module Stop 0   |
|     |          |                  |     | When this bit is set to 1, the clock supply to the UBC is halted. When this bit is cleared to 0, the clock supply to the UBC is restarted. The registers in the UBC are not initialized even when the clock supply to the UBC is halted.  |
|     |          |                  |     | 0: UBC runs   |
|     |          |                  |     | 1: Clock supply to UBC is halted  |
|     |          |                  |     |   |

## 29.3.2 Note on Accessing STBCR

The writing procedure for STBCR is different from that for other general registers to prevent it being modified easily. This register should be written to or read in the following procedure.

When writing to STBCR, be sure to use a word transfer instruction. Data cannot be written to by a byte transfer instruction. As shown in figure 29.1, transfer the data with the upper byte as H'3C and the lower byte as the write data. This transfer procedure writes the lower byte data to STBCR.

The reading procedure is similar to that for general registers. STBCR is allocated at address H'FFFE0400. Both byte transfer instructions and word transfer instructions are available.

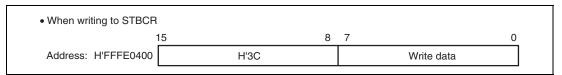


Figure 29.1 Writing to STBCR

#### **Operation** 29.4

#### 29.4.1 **Hardware Standby Mode**

#### **(1) Transition to Hardware Standby Mode**

This LSI enters hardware standby mode by setting the HSTBY pin low after setting the RES pin low. Set the mode pins as described in section 3, Operating Modes; otherwise, the correct operation is not guaranteed.

In hardware standby mode, all the functions of the LSI halt and the internal power supply to the area other than pages 0 and 1 of the on-chip RAM is halted, which reduces the LSI power consumption greatly. This LSI enters hardware standby mode asynchronously with and regardless of its current state because the transition is caused by an input on the external pin. Therefore, the LSI state before entering hardware standby mode is not retained except pages 0 and 1 of the onchip RAM.

The data stored in pages 0 and 1 of the on-chip RAM can be retained as long as the specific voltage is applied. To retain data stored in the on-chip RAM, clear the RAME0 bit (for page 0 area) and the RAME1 bit (for page 1 area) in the RAM enable control register (RAMEN) before setting the HSTBY pin low. For details on RAMEN, see section 28, RAM.

Hold the HSTBY pin low in hardware standby mode.

For the register states in hardware standby mode, see section 31, List of Registers. For the pin states in hardware standby mode, refer to appendix A, Pin States.

#### **Revoking Hardware Standby Mode (2)**

Hardware standby mode can be revoked only with the HSTBY and RES pins.

Setting the HSTBY pin high while the RES pin is low starts the clock oscillation. Be sure to hold the RES pin low until the clock oscillation is stabilized. After the clock oscillation has been stabilized, setting the RES pin high starts the power-on reset exception handling by the CPU.

#### (3)**Timing in Hardware Standby Mode**

Figure 29.2 shows an example of the timing of each pin in hardware standby mode.

This LSI enters hardware standby mode by setting the HSTBY pin low after setting the RES pin low. To revoke the mode, set the  $\overline{\text{HSTBY}}$  pin high, and after the clock has been stabilized set the RES pin high.

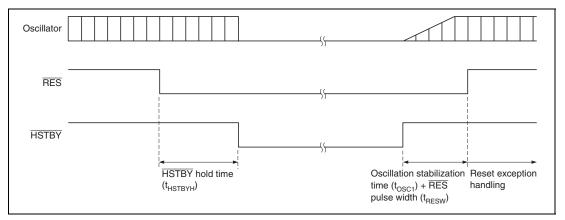


Figure 29.2 Timing in Hardware Standby Mode

## 29.4.2 Sleep Mode

## (1) Transition to Sleep Mode

Execution of the SLEEP instruction by the CPU causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run in sleep mode. Clock pulses continue to be output even in sleep mode if output from the CK pin is enabled by the CK control register (CKCR). For details on CKCR, refer to section 23, I/O Ports. For the state of each register in sleep mode, refer to section 31, List of Registers.

## (2) Revoking Sleep Mode

Sleep mode is revoked by an interrupt (NMI, IRQ, and on-chip peripheral module), DMA address error, or power-on reset.

## · Revoking with an interrupt

When an NMI, IRQ, or on-chip peripheral module interrupt occurs, sleep mode is revoked and interrupt exception handling is executed. When the priority level of the generated interrupt is equal to or lower than the interrupt mask level that is set in the status register (SR) of the CPU, or the interrupt by the on-chip peripheral module is disabled on the module side, the interrupt request is not accepted and sleep mode is not revoked.

- Revoking with a DMA address error
  - When a DMA address error occurs, sleep mode is revoked and DMA address error exception handling is executed.
- Revoking with a reset

When a low signal is input on the  $\overline{RES}$  pin or an internal reset by the WDT occurs, this LSI enters the power-on reset state and sleep mode is revoked.

## 29.4.3 Module Standby Function

## (1) Transition to Module Standby Function

Setting the MSTP bits in the standby control register to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This function can be used to reduce the power consumption in normal mode and sleep mode. Disable a module before placing it in the module standby mode. In addition, do not access the module's registers while it is in the module standby state.

The external pin states of the module in the module standby state vary. For details, see appendix A, Pin States.

The contents of registers in the UBC and the DMAC are retained even in the module standby state.

The JTAG TAP controller is put into a reset state (Test-Logic-Reset state) by the module standby function.

The internal state (including registers) of the AUD-II is initialized by the module standby function. However, the AUD-II does not enter the module standby state if the trace function is enabled (the EN bit in AUCSR is set to 1) while the AUD-II is operated in trace mode. Even if the trace function is disabled, the AUD-II does not enter in the module standby state as long as there remains the acquired trace data in the internal FIFO. The AUD-II switches to the module standby state after all remaining data has been output from the FIFO.

The FPU cannot restart operation once it has entered the module standby state. To restart the module, reset this LSI by a power-on reset.

## (2) Revoking Module Standby Function

The module standby function can be revoked by clearing the MSTP bits to 0, or by a power-on reset. When taking a module out of the module standby state by clearing the corresponding MSTP bit to 0, read the MSTP bit to confirm that it has been cleared to 0.

SH72546R Section 30 Reliability

## Section 30 Reliability

## 30.1 Reliability

A failure rate curve represents an index of the reliability of a semiconductor device. The failure rate curve traces a bathtub shape over the course of time, as is shown in figure 30.1. The curve is divided into three periods according to the type of failure phenomena: an initial failure period, a random failure period (functional lifetime), and a wear-out failure period. Initial failures, which occur during the initial failure period, are caused by contamination with foreign matter and localized chemical pollution; these can be eliminated by screening. Wear-out failures in the final period are caused by the deterioration of materials that make up semiconductor devices during long periods of usage. Random failures, which occur during the random failure period, are thought to occur in cases where a device with a minor failure is not removed by screening, and so is shipped, and then fails during the customer's production process or in the field, and in cases where a failure which should normally not have occurred until the wear-out period occurs earlier because of variations in production. Therefore, the reliability of semiconductor device is secured by appropriate screening to reduce the presence of initial failures and high reliability design to prevent the occurrence of wear-out failures. The reliability of a product is confirmed by producing a large quantity of prototypes for checking of the initial failure rate and executing accelerated life testing to identify the wear-out failure time in a realistic environment.

SH72546R Section 30 Reliability

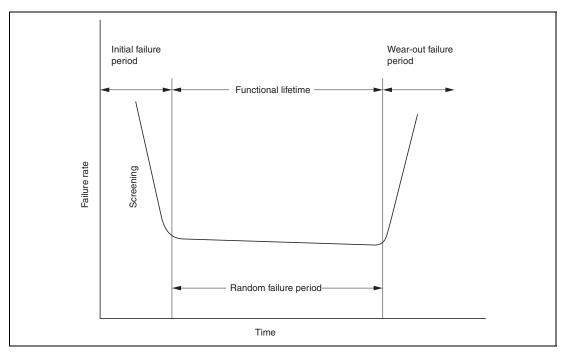


Figure 30.1 Failure Rate Curve (Bathtub Curve)

The reliability of products is estimated on the assumption that products developed for the automotive sector are used in a tougher environment than products for the consumer and industrial sectors. The representative failure phenomena of semiconductor devices, such as the dielectric breakdown of oxide films and electromigration in wiring, constitute wear-out failures. The stress factors in such failures are the voltage, current, and temperature applied to devices while they are in use. Since the temperature range for the guaranteed operation of products for use in automobiles is conventionally -40°C to 125°C, their reliability in terms of the above failure phenomena has to be confirmed by accelerated life testing at all temperatures in this range. Operation at temperatures in excess of 125°C leads to failure within a short time, since high temperatures induce failures in semiconductor devices.

## Section 31 List of Registers

The register list gives information on the on-chip I/O registers and is configured as described below.

## Register Addresses (grouped by module name, in ordered of the corresponding section numbers):

- Descriptions by functional module, in order of the corresponding section numbers
- Access to reserved addresses which are not described in this list is disabled.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given, on the presumption of a big-endian system.

## **List of Register Bits:**

- Bit configuration in each register is described in the same fashion as in Register Addresses (grouped by module name, in ordered of the corresponding section numbers).
- The character "-" in the bit name cell indicates a reserved bit.
- The blanks in the bit name cells in a whole line indicates that the corresponding register is allocated to a specific counter or data.

## **Register States in Each Operating Mode:**

- Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresponding section.
- The register states described are for the basic operating modes. If there is a specific reset for an on-chip module, refer to the section on that on-chip module.

# 31.1 Register Addresses (grouped by module name, in ordered of the corresponding section numbers)

| Module | Register Name                  | Abbreviation | Number of Bits | Address    | Access<br>Size |  |
|--------|--------------------------------|--------------|----------------|------------|----------------|--|
| INTC   | Interrupt control register 0   | ICR0         | 16             | H'FFFE0800 | 16, 32         |  |
|        | Interrupt control register 1   | ICR1         | 16             | H'FFFE0802 | 16, 32         |  |
|        | IRQ interrupt request register | IRQRR        | 16             | H'FFFE0806 | 8, 16, 32      |  |
|        | Bank control register          | IBCR         | 16             | H'FFFE080C | 16, 32         |  |
|        | Bank number register           | IBNR         | 16             | H'FFFE080E | 16, 32         |  |
|        | Software interrupt register 1  | SINTR1       | 8              | H'FFFE0810 | 8, 16, 32      |  |
|        | Software interrupt register 2  | SINTR2       | 8              | H'FFFE0811 | 8, 16, 32      |  |
|        | Software interrupt register 3  | SINTR3       | 8              | H'FFFE0812 | 8, 16, 32      |  |
|        | Software interrupt register 4  | SINTR4       | 8              | H'FFFE0813 | 8, 16, 32      |  |
|        | Software interrupt register 5  | SINTR5       | 8              | H'FFFE0814 | 8, 16, 32      |  |
|        | Software interrupt register 6  | SINTR6       | 8              | H'FFFE0815 | 8, 16, 32      |  |
|        | Software interrupt register 7  | SINTR7       | 8              | H'FFFE0816 | 8, 16, 32      |  |
|        | Software interrupt register 8  | SINTR8       | 8              | H'FFFE0817 | 8, 16, 32      |  |
|        | Interrupt priority register 01 | IPR01        | 16             | H'FFFE0818 | 16, 32         |  |
|        | Interrupt priority register 02 | IPR02        | 16             | H'FFFE081A | 16, 32         |  |
|        | Software interrupt register 9  | SINTR9       | 8              | H'FFFE0828 | 8, 16, 32      |  |
|        | Software interrupt register 10 | SINTR10      | 8              | H'FFFE0829 | 8, 16, 32      |  |
|        | Software interrupt register 11 | SINTR11      | 8              | H'FFFE082A | 8, 16, 32      |  |
|        | Software interrupt register 12 | SINTR12      | 8              | H'FFFE082B | 8, 16, 32      |  |
|        | Software interrupt register 13 | SINTR13      | 8              | H'FFFE082C | 8, 16, 32      |  |
|        | Software interrupt register 14 | SINTR14      | 8              | H'FFFE082D | 8, 16, 32      |  |
|        | Software interrupt register 15 | SINTR15      | 8              | H'FFFE082E | 8, 16, 32      |  |
|        | Interrupt priority register 03 | IPR03        | 16             | H'FFFE0C00 | 16, 32         |  |
|        | Interrupt priority register 04 | IPR04        | 16             | H'FFFE0C02 | 16, 32         |  |
|        | Interrupt priority register 05 | IPR05        | 16             | H'FFFE0C04 | 16, 32         |  |
|        | Interrupt priority register 06 | IPR06        | 16             | H'FFFE0C06 | 16, 32         |  |
|        | Interrupt priority register 07 | IPR07        | 16             | H'FFFE0C08 | 16, 32         |  |
|        | Interrupt priority register 08 | IPR08        | 16             | H'FFFE0C0A | 16, 32         |  |

| Module | Register Name                  | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|--------------------------------|--------------|----------------|------------|----------------|
| INTC   | Interrupt priority register 09 | IPR09        | 16             | H'FFFE0C0C | 16, 32         |
|        | Interrupt priority register 10 | IPR10        | 16             | H'FFFE0C0E | 16, 32         |
|        | Interrupt priority register 11 | IPR11        | 16             | H'FFFE0C10 | 16, 32         |
|        | Interrupt priority register 12 | IPR12        | 16             | H'FFFE0C12 | 16, 32         |
|        | Interrupt priority register 13 | IPR13        | 16             | H'FFFE0C14 | 16, 32         |
|        | Interrupt priority register 14 | IPR14        | 16             | H'FFFE0C16 | 16, 32         |
|        | Interrupt priority register 15 | IPR15        | 16             | H'FFFE0C18 | 16, 32         |
|        | Interrupt priority register 16 | IPR16        | 16             | H'FFFE0C1A | 16, 32         |
|        | Interrupt priority register 17 | IPR17        | 16             | H'FFFE0C1C | 16, 32         |
|        | Interrupt priority register 18 | IPR18        | 16             | H'FFFE0C1E | 16, 32         |
|        | Interrupt priority register 19 | IPR19        | 16             | H'FFFE0C20 | 16, 32         |
|        | Interrupt priority register 20 | IPR20        | 16             | H'FFFE0C22 | 16, 32         |
|        | Interrupt priority register 21 | IPR21        | 16             | H'FFFE0C24 | 16, 32         |
|        | Interrupt priority register 22 | IPR22        | 16             | H'FFFE0C26 | 16, 32         |
|        | Interrupt priority register 23 | IPR23        | 16             | H'FFFE0C28 | 16, 32         |
|        | Interrupt priority register 24 | IPR24        | 16             | H'FFFE0C2A | 16, 32         |
|        | Interrupt priority register 25 | IPR25        | 16             | H'FFFE0C2C | 16, 32         |
|        | Interrupt priority register 26 | IPR26        | 16             | H'FFFE0C2E | 16, 32         |
|        | Interrupt priority register 27 | IPR27        | 16             | H'FFFE0C30 | 16, 32         |
|        | Interrupt priority register 28 | IPR28        | 16             | H'FFFE0C32 | 16, 32         |
|        | Interrupt priority register 29 | IPR29        | 16             | H'FFFE0C34 | 16, 32         |
| UBC    | Break address register_0       | BAR_0        | 32             | H'FFFC0400 | 32             |
|        | Break address mask register_0  | BAMR_0       | 32             | H'FFFC0404 | 32             |
|        | Break bus cycle register_0     | BBR_0        | 16             | H'FFFC04A0 | 16             |
|        | Break address register_1       | BAR_1        | 32             | H'FFFC0410 | 32             |
|        | Break address mask register_1  | BAMR_1       | 32             | H'FFFC0414 | 32             |
|        | Break bus cycle register_1     | BBR_1        | 16             | H'FFFC04B0 | 16             |
|        | Break address register_2       | BAR_2        | 32             | H'FFFC0420 | 32             |
|        | Break address mask register_2  | BAMR_2       | 32             | H'FFFC0424 | 32             |
|        | Break bus cycle register_2     | BBR_2        | 16             | H'FFFC04A4 | 16             |
|        | Break address register_3       | BAR_3        | 32             | H'FFFC0430 | 32             |

| Module | Register Name                             | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|---|--------------|----------------|------------|----------------|
| UBC    | Break address mask register_3             | BAMR_3       | 32             | H'FFFC0434 | 32             |
|        | Break bus cycle register_3                | BBR_3        | 16             | H'FFFC04B4 | 16             |
|        | Break control register                    | BRCR         | 32             | H'FFFC04C0 | 8, 32          |
| BSC    | CS0 space bus control register            | CS0BCR       | 32             | H'FFFC0004 | 32             |
|        | CS1 space bus control register            | CS1BCR       | 32             | H'FFFC0008 | 32             |
|        | CS2 space bus control register            | CS2BCR       | 32             | H'FFFC000C | 32             |
|        | CS3 space bus control register            | CS3BCR       | 32             | H'FFFC0010 | 32             |
|        | CS0 space wait control register           | CS0WCR       | 32             | H'FFFC0028 | 32             |
|        | CS1 space wait control register           | CS1WCR       | 32             | H'FFFC002C | 32             |
|        | CS2 space wait control register           | CS2WCR       | 32             | H'FFFC0030 | 32             |
|        | CS3 space wait control register           | CS3WCR       | 32             | H'FFFC0034 | 32             |
| DMAC   | DMA source address register 0             | SAR0         | 32             | H'FFFE1000 | 16, 32         |
|        | DMA destination address register 0        | DAR0         | 32             | H'FFFE1004 | 16, 32         |
|        | DMA transfer count register 0             | DMATCR0      | 32             | H'FFFE1008 | 16, 32         |
|        | DMA channel control register 0            | CHCR0        | 32             | H'FFFE100C | 8, 16, 32      |
|        | DMA channel flag bit register 0           | CHFR0        | 8              | H'FFFE108C | 8              |
|        | DMA TE flag mask setting register 0       | TEMSK0       | 16             | H'FFFE108E | 8, 16          |
|        | DMA reload source address register 0      | RSAR0        | 32             | H'FFFE1100 | 16, 32         |
|        | DMA reload destination address register 0 | RDAR0        | 32             | H'FFFE1104 | 16, 32         |
|        | DMA reload transfer count register 0      | RDMATCR0     | 32             | H'FFFE1108 | 16, 32         |
|        | DMA source address register 1             | SAR1         | 32             | H'FFFE1010 | 16, 32         |
|        | DMA destination address register 1        | DAR1         | 32             | H'FFFE1014 | 16, 32         |
|        | DMA transfer count register 1             | DMATCR1      | 32             | H'FFFE1018 | 16, 32         |
|        | DMA channel control register 1            | CHCR1        | 32             | H'FFFE101C | 8, 16, 32      |
|        | DMA channel flag bit register 1           | CHFR1        | 8              | H'FFFE109C | 8              |
|        | DMA TE flag mask setting register 1       | TEMSK1       | 16             | H'FFFE109E | 8, 16          |

| Module | Register Name                             | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|---|--------------|----------------|------------|----------------|
| DMAC   | DMA reload source address register 1      | RSAR1        | 32             | H'FFFE1110 | 16, 32         |
|        | DMA reload destination address register 1 | RDAR1        | 32             | H'FFFE1114 | 16, 32         |
|        | DMA reload transfer count register 1      | RDMATCR1     | 32             | H'FFFE1118 | 16, 32         |
|        | DMA source address register 2             | SAR2         | 32             | H'FFFE1020 | 16, 32         |
|        | DMA destination address register 2        | DAR2         | 32             | H'FFFE1024 | 16, 32         |
|        | DMA transfer count register 2             | DMATCR2      | 32             | H'FFFE1028 | 16, 32         |
|        | DMA channel control register 2            | CHCR2        | 32             | H'FFFE102C | 8, 16, 32      |
|        | DMA channel flag bit register 2           | CHFR2        | 8              | H'FFFE10AC | 8              |
|        | DMA TE flag mask setting register 2       | TEMSK2       | 16             | H'FFFE10AE | 8, 16          |
|        | DMA reload source address register 2      | RSAR2        | 32             | H'FFFE1120 | 16, 32         |
|        | DMA reload destination address register 2 | RDAR2        | 32             | H'FFFE1124 | 16, 32         |
|        | DMA reload transfer count register 2      | RDMATCR2     | 32             | H'FFFE1128 | 16, 32         |
|        | DMA source address register 3             | SAR3         | 32             | H'FFFE1030 | 16, 32         |
|        | DMA destination address register 3        | DAR3         | 32             | H'FFFE1034 | 16, 32         |
|        | DMA transfer count register 3             | DMATCR3      | 32             | H'FFFE1038 | 16, 32         |
|        | DMA channel control register 3            | CHCR3        | 32             | H'FFFE103C | 8, 16, 32      |
|        | DMA channel flag bit register 3           | CHFR3        | 8              | H'FFFE10BC | 8              |
|        | DMA TE flag mask setting register 3       | TEMSK3       | 16             | H'FFFE10BE | 8, 16          |
|        | DMA reload source address register 3      | RSAR3        | 32             | H'FFFE1130 | 16, 32         |
|        | DMA reload destination address register 3 | RDAR3        | 32             | H'FFFE1134 | 16, 32         |
|        | DMA reload transfer count register 3      | RDMATCR3     | 32             | H'FFFE1138 | 16, 32         |
|        | DMA source address register 4             | SAR4         | 32             | H'FFFE1040 | 16, 32         |

| Module | Register Name                              | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|--|--------------|----------------|------------|----------------|
| DMAC   | DMA destination address register 4         | DAR4         | 32             | H'FFFE1044 | 16, 32         |
|        | DMA transfer count register 4              | DMATCR4      | 32             | H'FFFE1048 | 16, 32         |
|        | DMA channel control register 4             | CHCR4        | 32             | H'FFFE104C | 8, 16, 32      |
|        | DMA channel flag bit register 4            | CHFR4        | 8              | H'FFFE10CC | 8              |
|        | DMA TE flag mask setting register 4        | TEMSK4       | 16             | H'FFFE10CE | 8, 16          |
|        | DMA reload source address register 4       | RSAR4        | 32             | H'FFFE1140 | 16, 32         |
|        | DMA reload destination address register 4  | RDAR4        | 32             | H'FFFE1144 | 16, 32         |
|        | DMA reload transfer count register 4       | RDMATCR4     | 32             | H'FFFE1148 | 16, 32         |
|        | DMA address reload count register 4        | ARCR4        | 16             | H'FFFE114C | 16, 32         |
|        | DMA reload-address reload count register 4 | RARCR4       | 16             | H'FFFE114E | 16             |
|        | DMA source address register 5              | SAR5         | 32             | H'FFFE1050 | 16, 32         |
|        | DMA destination address register 5         | DAR5         | 32             | H'FFFE1054 | 16, 32         |
|        | DMA transfer count register 5              | DMATCR5      | 32             | H'FFFE1058 | 16, 32         |
|        | DMA channel control register 5             | CHCR5        | 32             | H'FFFE105C | 8, 16, 32      |
|        | DMA channel flag bit register 5            | CHFR5        | 8              | H'FFFE10DC | 8              |
|        | DMA TE flag mask setting register 5        | TEMSK5       | 16             | H'FFFE10DE | 8, 16          |
|        | DMA reload source address register 5       | RSAR5        | 32             | H'FFFE1150 | 16, 32         |
|        | DMA reload destination address register 5  | RDAR5        | 32             | H'FFFE1154 | 16, 32         |
|        | DMA reload transfer count register 5       | RDMATCR5     | 32             | H'FFFE1158 | 16, 32         |
|        | DMA address reload count register 5        | ARCR5        | 16             | H'FFFE115C | 16, 32         |
|        | DMA reload-address reload count register 5 | RARCR5       | 16             | H'FFFE115E | 16             |

| Module | Register Name                              | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|--|--------------|----------------|------------|----------------|
| DMAC   | DMA source address register 6              | SAR6         | 32             | H'FFFE1060 | 16, 32         |
|        | DMA destination address register 6         | DAR6         | 32             | H'FFFE1064 | 16, 32         |
|        | DMA transfer count register 6              | DMATCR6      | 32             | H'FFFE1068 | 16, 32         |
|        | DMA channel control register 6             | CHCR6        | 32             | H'FFFE106C | 8, 16, 32      |
|        | DMA channel flag bit register 6            | CHFR6        | 8              | H'FFFE10EC | 8              |
|        | DMA TE flag mask setting register 6        | TEMSK6       | 16             | H'FFFE10EE | 8, 16          |
|        | DMA reload source address register 6       | RSAR6        | 32             | H'FFFE1160 | 16, 32         |
|        | DMA reload destination address register 6  | RDAR6        | 32             | H'FFFE1164 | 16, 32         |
|        | DMA reload transfer count register 6       | RDMATCR6     | 32             | H'FFFE1168 | 16, 32         |
|        | DMA address reload count register 6        | ARCR6        | 16             | H'FFFE116C | 16, 32         |
|        | DMA reload-address reload count register 6 | RARCR6       | 16             | H'FFFE116E | 16             |
|        | DMA source address register 7              | SAR7         | 32             | H'FFFE1070 | 16, 32         |
|        | DMA destination address register 7         | DAR7         | 32             | H'FFFE1074 | 16, 32         |
|        | DMA transfer count register 7              | DMATCR7      | 32             | H'FFFE1078 | 16, 32         |
|        | DMA channel control register 7             | CHCR7        | 32             | H'FFFE107C | 8, 16, 32      |
|        | DMA channel flag bit register 7            | CHFR7        | 8              | H'FFFE10FC | 8              |
|        | DMA TE flag mask setting register 7        | TEMSK7       | 16             | H'FFFE10FE | 8, 16          |
|        | DMA reload source address register 7       | RSAR7        | 32             | H'FFFE1170 | 16, 32         |
|        | DMA reload destination address register_7  | RDAR7        | 32             | H'FFFE1174 | 16, 32         |
|        | DMA reload transfer count register 7       | RDMATCR7     | 32             | H'FFFE1178 | 16, 32         |
|        | DMA address reload count register 7        | ARCR7        | 16             | H'FFFE117C | 16, 32         |
|        | DMA reload-address reload count register 7 | RARCR7       | 16             | H'FFFE117E | 16             |

| Module | Register Name                       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|-------------------------------------|--------------|----------------|------------|----------------|
| DMAC   | DMA operation register              | DMAOR        | 16             | H'FFFE1200 | 8, 16          |
|        | DMA operation flag bit register     | DMAFR        | 8              | H'FFFE1204 | 8              |
|        | DMA extension resource selector 0   | DMARS0       | 16             | H'FFFE1300 | 8, 16          |
|        | DMA extension resource selector 1   | DMARS1       | 16             | H'FFFE1304 | 8, 16          |
|        | DMA extension resource selector 2   | DMARS2       | 16             | H'FFFE1308 | 8, 16          |
|        | DMA extension resource selector 3   | DMARS3       | 16             | H'FFFE130C | 8, 16          |
| A-DMAC | A-DMAC operation register           | ADMAOR       | 8              | H'FFFE6000 | 8              |
|        | A-DMAC alias base register          | ADMAABR      | 8              | H'FFFE6002 | 8              |
|        | A-DMAC interrupt control register 0 | ADMAIE0      | 8              | H'FFFE6010 | 8              |
|        | A-DMAC interrupt control register 1 | ADMAIE1      | 8              | H'FFFE6011 | 8              |
|        | A-DMAC interrupt control register 2 | ADMAIE2      | 8              | H'FFFE6012 | 8              |
|        | A-DMAC interrupt control register 3 | ADMAIE3      | 8              | H'FFFE6013 | 8              |
|        | A-DMAC interrupt control register 4 | ADMAIE4      | 8              | H'FFFE6014 | 8              |
|        | A-DMAC interrupt control register 5 | ADMAIE5      | 8              | H'FFFE6015 | 8              |
|        | A-DMAC interrupt control register 7 | ADMAIE7      | 8              | H'FFFE6017 | 8              |
|        | A-DMAC interrupt control register 8 | ADMAIE8      | 8              | H'FFFE6018 | 8              |
|        | A-DMAC interrupt control register 9 | ADMAIE9      | 8              | H'FFFE6019 | 8              |
|        | A-DMAC data valid register 0        | ADMADV0      | 8              | H'FFFE6020 | 8              |
|        | A-DMAC data valid register 1        | ADMADV1      | 8              | H'FFFE6021 | 8              |
|        | A-DMAC data valid register 2        | ADMADV2      | 8              | H'FFFE6022 | 8              |
|        | A-DMAC data valid register 3        | ADMADV3      | 8              | H'FFFE6023 | 8              |
|        | A-DMAC data valid register 4        | ADMADV4      | 8              | H'FFFE6024 | 8              |

| Module | Register Name                           | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|---|--------------|----------------|------------|----------------|
| A-DMAC | A-DMAC data valid register 5            | ADMADV5      | 8              | H'FFFE6025 | 8              |
|        | A-DMAC transfer end register 0          | ADMATE0      | 8              | H'FFFE6030 | 8              |
|        | A-DMAC transfer end register 1          | ADMATE1      | 8              | H'FFFE6031 | 8              |
|        | A-DMAC transfer end register 2          | ADMATE2      | 8              | H'FFFE6032 | 8              |
|        | A-DMAC enable register 0                | ADMADE0      | 8              | H'FFFE6040 | 8              |
|        | A-DMAC enable register 1                | ADMADE1      | 8              | H'FFFE6041 | 8              |
|        | A-DMAC enable register 2                | ADMADE2      | 8              | H'FFFE6042 | 8              |
|        | A-DMAC enable register 3                | ADMADE3      | 8              | H'FFFE6043 | 8              |
|        | A-DMAC enable register 4                | ADMADE4      | 8              | H'FFFE6044 | 8              |
|        | A-DMAC enable register 5                | ADMADE5      | 8              | H'FFFE6045 | 8              |
|        | A-DMAC enable register 7                | ADMADE7      | 8              | H'FFFE6047 | 8              |
|        | A-DMAC transfer mode register 0         | ADMAMODE0    | 8              | H'FFFE6050 | 8              |
|        | A-DMAC transfer mode register 1         | ADMAMODE1    | 8              | H'FFFE6051 | 8              |
|        | A-DMAC transfer mode register 2         | ADMAMODE2    | 8              | H'FFFE6052 | 8              |
|        | A-DMAC transfer count register 0        | ADMATCR0     | 16             | H'FFFE6060 | 16             |
|        | A-DMAC reload transfer count register 0 | ADMARTCR0    | 16             | H'FFFE6062 | 16             |
|        | A-DMAC transfer count register 1        | ADMATCR1     | 16             | H'FFFE6064 | 16             |
|        | A-DMAC reload transfer count register 1 | ADMARTCR1    | 16             | H'FFFE6066 | 16             |
|        | A-DMAC transfer count register 56       | ADMATCR56    | 16             | H'FFFE6070 | 16             |
|        | A-DMAC transfer count register 57       | ADMATCR57    | 16             | H'FFFE6072 | 16             |
|        | A-DMAC transfer count register 58       | ADMATCR58    | 16             | H'FFFE6074 | 16             |
|        | A-DMAC transfer count register 59       | ADMATCR59    | 16             | H'FFFE6076 | 16             |
|        | A-DMAC transfer count register 60       | ADMATCR60    | 16             | H'FFFE6078 | 16             |
|        | A-DMAC transfer count register 61       | ADMATCR61    | 16             | H'FFFE607A | 16             |
|        | A-DMAC transfer count register 62       | ADMATCR62    | 16             | H'FFFE607C | 16             |

| Module | Register Name                          | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|--|--------------|----------------|------------|----------------|
| A-DMAC | A-DMAC transfer count register 63      | ADMATCR63    | 16             | H'FFFE607E | 16             |
|        | A-DMAC transfer count register 64      | ADMATCR64    | 16             | H'FFFE6080 | 16             |
|        | A-DMAC transfer count register 65      | ADMATCR65    | 16             | H'FFFE6082 | 16             |
|        | A-DMAC transfer count register 66      | ADMATCR66    | 16             | H'FFFE6084 | 16             |
|        | A-DMAC transfer count register 67      | ADMATCR67    | 16             | H'FFFE6086 | 16             |
|        | A-DMAC transfer count register 68      | ADMATCR68    | 16             | H'FFFE6088 | 16             |
|        | A-DMAC transfer count register 69      | ADMATCR69    | 16             | H'FFFE608A | 16             |
|        | A-DMAC transfer count register 70      | ADMATCR70    | 16             | H'FFFE608C | 16             |
|        | A-DMAC transfer count register 71      | ADMATCR71    | 16             | H'FFFE608E | 16             |
|        | A-DMAC alias pointer register 0        | ADMAAR0      | 16             | H'FFFE6090 | 16             |
|        | A-DMAC reload alias pointer register 0 | ADMARAR0     | 16             | H'FFFE6092 | 16             |
|        | A-DMAC alias pointer register 1        | ADMAAR1      | 16             | H'FFFE6094 | 16             |
|        | A-DMAC reload alias pointer register 1 | ADMARAR1     | 16             | H'FFFE6096 | 16             |
|        | A-DMAC alias pointer register 56       | ADMAAR56     | 16             | H'FFFE60A0 | 16             |
|        | A-DMAC alias pointer register 57       | ADMAAR57     | 16             | H'FFFE60A2 | 16             |
|        | A-DMAC alias pointer register 58       | ADMAAR58     | 16             | H'FFFE60A4 | 16             |
|        | A-DMAC alias pointer register 59       | ADMAAR59     | 16             | H'FFFE60A6 | 16             |
|        | A-DMAC alias pointer register 60       | ADMAAR60     | 16             | H'FFFE60A8 | 16             |
|        | A-DMAC alias pointer register 61       | ADMAAR61     | 16             | H'FFFE60AA | 16             |
|        | A-DMAC alias pointer register 62       | ADMAAR62     | 16             | H'FFFE60AC | 16             |
|        | A-DMAC alias pointer register 63       | ADMAAR63     | 16             | H'FFFE60AE | 16             |
|        | A-DMAC alias pointer register 64       | ADMAAR64     | 16             | H'FFFE60B0 | 16             |
|        | A-DMAC alias pointer register 65       | ADMAAR65     | 16             | H'FFFE60B2 | 16             |

| Module  | Register Name                    | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|----------------------------------|--------------|----------------|------------|----------------|
| A-DMAC  | A-DMAC alias pointer register 66 | ADMAAR66     | 16             | H'FFFE60B4 | 16             |
|         | A-DMAC alias pointer register 67 | ADMAAR67     | 16             | H'FFFE60B6 | 16             |
|         | A-DMAC alias pointer register 68 | ADMAAR68     | 16             | H'FFFE60B8 | 16             |
|         | A-DMAC alias pointer register 69 | ADMAAR69     | 16             | H'FFFE60BA | 16             |
|         | A-DMAC alias pointer register 70 | ADMAAR70     | 16             | H'FFFE60BC | 16             |
|         | A-DMAC alias pointer register 71 | ADMAAR71     | 16             | H'FFFE60BE | 16             |
|         | A-DMAC buffer register 2         | ADMABUF2     | 32             | H'FFFE60C0 | 32             |
|         | A-DMAC buffer register 3         | ADMABUF3     | 32             | H'FFFE60C4 | 32             |
|         | A-DMAC buffer register 4         | ADMABUF4     | 32             | H'FFFE60C8 | 32             |
|         | A-DMAC buffer register 5         | ADMABUF5     | 32             | H'FFFE60CC | 32             |
|         | A-DMAC buffer register 6         | ADMABUF6     | 32             | H'FFFE60D0 | 32             |
|         | A-DMAC buffer register 7         | ADMABUF7     | 32             | H'FFFE60D4 | 32             |
|         | A-DMAC receive wait register 0   | ADMARVPR0    | 16             | H'FFFE60E0 | 8, 16          |
|         | A-DMAC receive wait register 1   | ADMARVPR1    | 16             | H'FFFE60E2 | 8, 16          |
|         | A-DMAC receive wait register 2   | ADMARVPR2    | 16             | H'FFFE60E4 | 8, 16          |
|         | A-DMAC receive wait register 3   | ADMARVPR3    | 16             | H'FFFE60E6 | 8, 16          |
|         | A-DMAC receive wait register 4   | ADMARVPR4    | 16             | H'FFFE60E8 | 8, 16          |
|         | A-DMAC receive wait register 5   | ADMARVPR5    | 16             | H'FFFE60EA | 8, 16          |
|         | A-DMAC transmit wait register 0  | ADMATVPR0    | 16             | H'FFFE60F0 | 8, 16          |
|         | A-DMAC transmit wait register 1  | ADMATVPR1    | 16             | H'FFFE60F2 | 8, 16          |
|         | A-DMAC transmit wait register 2  | ADMATVPR2    | 16             | H'FFFE60F4 | 8, 16          |
|         | A-DMAC transmit wait register 3  | ADMATVPR3    | 16             | H'FFFE60F6 | 8, 16          |
|         | A-DMAC transmit wait register 4  | ADMATVPR4    | 16             | H'FFFE60F8 | 8, 16          |
|         | A-DMAC transmit wait register 5  | ADMATVPR5    | 16             | H'FFFE60FA | 8, 16          |
| ATU-III | ATU-III master enable register   | ATUENR       | 16             | H'FFFFF000 | 8, 16          |
|         | Clock bus control register       | CBCNT        | 8              | H'FFFFF002 | 8              |
|         | Noise cancellation mode register | NCMR         | 8              | H'FFFFF003 | 8              |
|         | Prescaler register 0             | PSCR0        | 16             | H'FFFFF100 | 16             |
|         | Prescaler register 1             | PSCR1        | 16             | H'FFFFF102 | 16             |
|         | Prescaler register 2             | PSCR2        | 16             | H'FFFFF104 | 16             |
|         | Prescaler register 3             | PSCR3        | 16             | H'FFFFF106 | 16             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Timer control register A           | TCRA         | 8              | H'FFFFF202 | 8              |
|         | Timer I/O control register 1A      | TIOR1A       | 16             | H'FFFFF204 | 8, 16          |
|         | Timer I/O control register 2A      | TIOR2A       | 16             | H'FFFFF206 | 8, 16          |
|         | Timer status register A            | TSRA         | 8              | H'FFFFF208 | 8              |
|         | Timer interrupt enable register A  | TIERA        | 8              | H'FFFFF209 | 8              |
|         | Noise canceler counter A0          | NCNTA0       | 8              | H'FFFFF210 | 8              |
|         | Noise canceler register A0         | NCRA0        | 8              | H'FFFFF211 | 8              |
|         | Noise canceler counter A1          | NCNTA1       | 8              | H'FFFFF212 | 8              |
|         | Noise canceler register A1         | NCRA1        | 8              | H'FFFFF213 | 8              |
|         | Noise canceler counter A2          | NCNTA2       | 8              | H'FFFFF214 | 8              |
|         | Noise canceler register A2         | NCRA2        | 8              | H'FFFFF215 | 8              |
|         | Noise canceler counter A3          | NCNTA3       | 8              | H'FFFFF216 | 8              |
|         | Noise canceler register A3         | NCRA3        | 8              | H'FFFFF217 | 8              |
|         | Noise canceler counter A4          | NCNTA4       | 8              | H'FFFFF218 | 8              |
|         | Noise canceler register A4         | NCRA4        | 8              | H'FFFFF219 | 8              |
|         | Noise canceler counter A5          | NCNTA5       | 8              | H'FFFFF21A | 8              |
|         | Noise canceler register A5         | NCRA5        | 8              | H'FFFFF21B | 8              |
|         | Free-running counter A             | TCNTA        | 32             | H'FFFFF220 | 32             |
|         | Input capture register A0          | ICRA0        | 32             | H'FFFFF228 | 32             |
|         | Input capture register A1          | ICRA1        | 32             | H'FFFFF22C | 32             |
|         | Input capture register A2          | ICRA2        | 32             | H'FFFFF230 | 32             |
|         | Input capture register A3          | ICRA3        | 32             | H'FFFFF234 | 32             |
|         | Input capture register A4          | ICRA4        | 32             | H'FFFFF238 | 32             |
|         | Input capture register A5          | ICRA5        | 32             | H'FFFFF23C | 32             |
|         | Timer control register B           | TCRB         | 8              | H'FFFFF304 | 8              |
|         | Timer I/O control register B       | TIORB        | 8              | H'FFFFF305 | 8              |
|         | Timer status register B            | TSRB         | 8              | H'FFFFF306 | 8              |
|         | Timer interrupt enable register B  | TIERB        | 8              | H'FFFFF307 | 8              |
|         | Edge interval measuring counter B0 | TCNTB0       | 32             | H'FFFFF310 | 32             |
|         | Input capture register B0          | ICRB0        | 32             | H'FFFFF314 | 32             |

| Module  | Register Name  | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|--|--------------|----------------|------------|----------------|
| ATU-III | Output compare register B0                                 | OCRB0        | 32             | H'FFFFF318 | 32             |
|         | Event counter B1   | TCNTB1       | 8              | H'FFFFF31C | 8              |
|         | Output compare register B1                                 | OCRB1        | 8              | H'FFFFF31D | 8              |
|         | Input capture register B1                                  | ICRB1        | 32             | H'FFFFF320 | 32             |
|         | Input capture register B2                                  | ICRB2        | 32             | H'FFFFF324 | 32             |
|         | Load register B  | LDB          | 32             | H'FFFFF330 | 32             |
|         | Reload register B  | RLDB         | 32             | H'FFFFF334 | 32             |
|         | Pulse interval multiplier register                         | PIMR         | 16             | H'FFFFF338 | 16             |
|         | Reloadable counter B2                                      | TCNTB2       | 32             | H'FFFFF33C | 32             |
|         | Multiplied clock counter B6                                | TCNTB6       | 32             | H'FFFFF340 | 32             |
|         | Output compare register B6                                 | OCRB6        | 32             | H'FFFFF344 | 32             |
|         | Output compare register B7                                 | OCRB7        | 32             | H'FFFFF348 | 32             |
|         | Correcting event counter B3                                | TCNTB3       | 32             | H'FFFFF350 | 32             |
|         | Multiplied-and-corrected clock counter B4                  | TCNTB4       | 32             | H'FFFFF354 | 32             |
|         | Generating frequency-multiplied corrected clock counter B5 | TCNTB5       | 32             | H'FFFFF358 | 32             |
|         | Correcting counter clearing register B                     | TCCLRB       | 32             | H'FFFFF35C | 32             |
|         | Timer start register C                                     | TSTRC        | 8              | H'FFFFF400 | 8              |
|         | Noise canceler control register C0                         | NCCRC0       | 8              | H'FFFFF402 | 8              |
|         | Noise canceler control register C1                         | NCCRC1       | 8              | H'FFFFF403 | 8              |
|         | Noise canceler control register C2                         | NCCRC2       | 8              | H'FFFFF404 | 8              |
|         | Noise canceler control register C3                         | NCCRC3       | 8              | H'FFFFF405 | 8              |
|         | Noise canceler control register C4                         | NCCRC4       | 8              | H'FFFFF406 | 8              |
|         | Noise canceler counter C00                                 | NCNTC00      | 8              | H'FFFFF410 | 8              |
|         | Noise canceler counter C01                                 | NCNTC01      | 8              | H'FFFFF411 | 8              |
|         | Noise canceler counter C02                                 | NCNTC02      | 8              | H'FFFFF412 | 8              |
|         | Noise canceler counter C03                                 | NCNTC03      | 8              | H'FFFFF413 | 8              |

| Noise car<br>Noise car<br>Noise car | ncel register C00 ncel register C01 ncel register C02 ncel register C03 nceler counter C10 | NCRC00<br>NCRC01<br>NCRC02<br>NCRC03 | 8 8 | H'FFFFF414<br>H'FFFFF415 | 8 |
|-------------------------------------|--|--------------------------------------|-----|--------------------------|---|
| Noise car<br>Noise car              | ncel register C02  | NCRC02                               |     | H'FFFFF415               | 0 |
| Noise car                           | ncel register C03  |                                      | 8   |                          | 8 |
| Noise car                           |  | NCRC03                               |     | H'FFFFF416               | 8 |
|                                     | nceler counter C10   |                                      | 8   | H'FFFFF417               | 8 |
| Noise car                           |  | NCNTC10                              | 8   | H'FFFFF418               | 8 |
|                                     | nceler counter C11   | NCNTC11                              | 8   | H'FFFFF419               | 8 |
| Noise car                           | nceler counter C12   | NCNTC12                              | 8   | H'FFFFF41A               | 8 |
| Noise car                           | nceler counter C13   | NCNTC13                              | 8   | H'FFFFF41B               | 8 |
| Noise car                           | ncel register C10  | NCRC10                               | 8   | H'FFFFF41C               | 8 |
| Noise car                           | ncel register C11  | NCRC11                               | 8   | H'FFFFF41D               | 8 |
| Noise car                           | ncel register C12  | NCRC12                               | 8   | H'FFFFF41E               | 8 |
| Noise car                           | ncel register C13  | NCRC13                               | 8   | H'FFFFF41F               | 8 |
| Noise car                           | nceler counter C20   | NCNTC20                              | 8   | H'FFFFF420               | 8 |
| Noise car                           | nceler counter C21   | NCNTC21                              | 8   | H'FFFFF421               | 8 |
| Noise car                           | nceler counter C22   | NCNTC22                              | 8   | H'FFFFF422               | 8 |
| Noise car                           | nceler counter C23   | NCNTC23                              | 8   | H'FFFFF423               | 8 |
| Noise car                           | ncel register C20  | NCRC20                               | 8   | H'FFFFF424               | 8 |
| Noise car                           | ncel register C21  | NCRC21                               | 8   | H'FFFFF425               | 8 |
| Noise car                           | ncel register C22  | NCRC22                               | 8   | H'FFFFF426               | 8 |
| Noise car                           | ncel register C23  | NCRC23                               | 8   | H'FFFFF427               | 8 |
| Noise car                           | nceler counter C30   | NCNTC30                              | 8   | H'FFFFF428               | 8 |
| Noise car                           | nceler counter C31   | NCNTC31                              | 8   | H'FFFFF429               | 8 |
| Noise car                           | nceler counter C32   | NCNTC32                              | 8   | H'FFFFF42A               | 8 |
| Noise car                           | nceler counter C33   | NCNTC33                              | 8   | H'FFFFF42B               | 8 |
| Noise car                           | ncel register C30  | NCRC30                               | 8   | H'FFFFF42C               | 8 |
| Noise car                           | ncel register C31  | NCRC31                               | 8   | H'FFFFF42D               | 8 |
| Noise car                           | ncel register C32  | NCRC32                               | 8   | H'FFFFF42E               | 8 |
| Noise car                           | ncel register C33  | NCRC33                               | 8   | H'FFFFF42F               | 8 |
| Noise car                           | nceler counter C40   | NCNTC40                              | 8   | H'FFFFF430               | 8 |
| Noise car                           | nceler counter C41   | NCNTC41                              | 8   | H'FFFFF431               | 8 |
| Noise car                           | nceler counter C42   | NCNTC42                              | 8   | H'FFFFF432               | 8 |
| Noise car                           | nceler counter C43   | NCNTC43                              | 8   | H'FFFFF433               | 8 |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Noise cancel register C40          | NCRC40       | 8              | H'FFFFF434 | 8              |
|         | Noise cancel register C41          | NCRC41       | 8              | H'FFFFF435 | 8              |
|         | Noise cancel register C42          | NCRC42       | 8              | H'FFFFF436 | 8              |
|         | Noise cancel register C43          | NCRC43       | 8              | H'FFFFF437 | 8              |
|         | Timer control register C0          | TCRC0        | 8              | H'FFFFF440 | 8              |
|         | Timer interrupt enable register C0 | TIERC0       | 8              | H'FFFFF441 | 8              |
|         | Timer I/O control register C0      | TIORC0       | 16             | H'FFFFF442 | 8, 16          |
|         | Timer status register C0           | TSRC0        | 8              | H'FFFFF444 | 8              |
|         | General register C00               | GRC00        | 32             | H'FFFFF448 | 32             |
|         | General register C01               | GRC01        | 32             | H'FFFFF44C | 32             |
|         | General register C02               | GRC02        | 32             | H'FFFFF450 | 32             |
|         | General register C03               | GRC03        | 32             | H'FFFFF454 | 32             |
|         | Timer counter C0                   | TCNTC0       | 32             | H'FFFFF458 | 32             |
|         | Timer control register C1          | TCRC1        | 8              | H'FFFFF460 | 8              |
|         | Timer interrupt enable register C1 | TIERC1       | 8              | H'FFFFF461 | 8              |
|         | Timer I/O control register C1      | TIORC1       | 16             | H'FFFFF462 | 8, 16          |
|         | Timer status register C1           | TSRC1        | 8              | H'FFFFF464 | 8              |
|         | General register C10               | GRC10        | 32             | H'FFFFF468 | 32             |
|         | General register C11               | GRC11        | 32             | H'FFFFF46C | 32             |
|         | General register C12               | GRC12        | 32             | H'FFFFF470 | 32             |
|         | General register C13               | GRC13        | 32             | H'FFFFF474 | 32             |
|         | Timer counter C1                   | TCNTC1       | 32             | H'FFFFF478 | 32             |
|         | Timer control register C2          | TCRC2        | 8              | H'FFFFF480 | 8              |
|         | Timer interrupt enable register C2 | TIERC2       | 8              | H'FFFFF481 | 8              |
|         | Timer I/O control register C2      | TIORC2       | 16             | H'FFFFF482 | 8, 16          |
|         | Timer status register C2           | TSRC2        | 8              | H'FFFFF484 | 8              |
|         | General register C20               | GRC20        | 32             | H'FFFFF488 | 32             |
|         | General register C21               | GRC21        | 32             | H'FFFFF48C | 32             |
|         | General register C22               | GRC22        | 32             | H'FFFFF490 | 32             |

| Module  | Register Name                                  | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|--|--------------|----------------|------------|----------------|
| ATU-III | General register C23                           | GRC23        | 32             | H'FFFFF494 | 32             |
|         | Timer counter C2                               | TCNTC2       | 32             | H'FFFFF498 | 32             |
|         | Timer control register C3                      | TCRC3        | 8              | H'FFFFF4A0 | 8              |
|         | Timer interrupt enable register C3             | TIERC3       | 8              | H'FFFFF4A1 | 8              |
|         | Timer I/O control register C3                  | TIORC3       | 16             | H'FFFFF4A2 | 8, 16          |
|         | Timer status register C3                       | TSRC3        | 8              | H'FFFFF4A4 | 8              |
|         | General register C30                           | GRC30        | 32             | H'FFFFF4A8 | 32             |
|         | General register C31                           | GRC31        | 32             | H'FFFFF4AC | 32             |
|         | General register C32                           | GRC32        | 32             | H'FFFFF4B0 | 32             |
|         | General register C33                           | GRC33        | 32             | H'FFFFF4B4 | 32             |
|         | Timer counter C3                               | TCNTC3       | 32             | H'FFFFF4B8 | 32             |
|         | Timer control register C4                      | TCRC4        | 8              | H'FFFFF4C0 | 8              |
|         | Timer interrupt enable register C4             | TIERC4       | 8              | H'FFFFF4C1 | 8              |
|         | Timer I/O control register C4                  | TIORC4       | 16             | H'FFFFF4C2 | 8, 16          |
|         | Timer status register C4                       | TSRC4        | 8              | H'FFFFF4C4 | 8              |
|         | General register C40                           | GRC40        | 32             | H'FFFFF4C8 | 32             |
|         | General register C41                           | GRC41        | 32             | H'FFFFF4CC | 32             |
|         | General register C42                           | GRC42        | 32             | H'FFFFF4D0 | 32             |
|         | General register C43                           | GRC43        | 32             | H'FFFFF4D4 | 32             |
|         | Timer counter C4                               | TCNTC4       | 32             | H'FFFFF4D8 | 32             |
|         | Timer start register                           | TSTRD        | 8              | H'FFFFF500 | 8              |
|         | Timer counter 1D0                              | TCNT1D0      | 32             | H'FFFFF520 | 32             |
|         | Timer counter 2D0                              | TCNT2D0      | 32             | H'FFFFF524 | 32             |
|         | Timer offset base register D0                  | OSBRD0       | 32             | H'FFFFF528 | 32             |
|         | Timer control register D0                      | TCRD0        | 16             | H'FFFFF52C | 8, 16          |
|         | Timer output control register D0               | TOCRD0       | 8              | H'FFFFF52E | 8              |
|         | Compare match pulse output control register D0 | CMPOD0       | 8              | H'FFFFF52F | 8              |
|         | Timer counter 1D1                              | TCNT1D1      | 32             | H'FFFFF530 | 32             |
|         | Timer counter 2D1                              | TCNT2D1      | 32             | H'FFFFF534 | 32             |

| Module  | Register Name                                  | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|--|--------------|----------------|------------|----------------|
| ATU-III | Timer offset base register D1                  | OSBRD1       | 32             | H'FFFFF538 | 32             |
|         | Timer control register D1                      | TCRD1        | 16             | H'FFFFF53C | 8, 16          |
|         | Timer output control register D1               | TOCRD1       | 8              | H'FFFFF53E | 8              |
|         | Compare match pulse output control register D1 | CMPOD1       | 8              | H'FFFFF53F | 8              |
|         | Timer counter 1D2                              | TCNT1D2      | 32             | H'FFFFF540 | 32             |
|         | Timer counter 2D2                              | TCNT2D2      | 32             | H'FFFFF544 | 32             |
|         | Timer offset base register D2                  | OSBRD2       | 32             | H'FFFFF548 | 32             |
|         | Timer control register D2                      | TCRD2        | 16             | H'FFFFF54C | 8, 16          |
|         | Timer output control register D2               | TOCRD2       | 8              | H'FFFFF54E | 8              |
|         | Timer counter 1D3                              | TCNT1D3      | 32             | H'FFFFF550 | 32             |
|         | Timer counter 2D3                              | TCNT2D3      | 32             | H'FFFFF554 | 32             |
|         | Timer offset base register D3                  | OSBRD3       | 32             | H'FFFFF558 | 32             |
|         | Timer control register D3                      | TCRD3        | 16             | H'FFFFF55C | 8, 16          |
|         | Timer output control register D3               | TOCRD3       | 8              | H'FFFFF55E | 8              |
|         | Timer I/O control register 1D0                 | TIOR1D0      | 16             | H'FFFFF580 | 8, 16          |
|         | Timer I/O control register 2D0                 | TIOR2D0      | 16             | H'FFFFF582 | 8, 16          |
|         | Down counter starting register D0              | DSTRD0       | 8              | H'FFFFF585 | 8              |
|         | Down counter status register D0                | DSRD0        | 8              | H'FFFFF587 | 8              |
|         | Down counter control register D0               | DCRD0        | 16             | H'FFFFF588 | 8, 16          |
|         | Timer status register D0                       | TSRD0        | 16             | H'FFFFF58C | 8, 16          |
|         | Timer interrupt enable register D0             | TIERD0       | 16             | H'FFFFF58E | 8, 16          |
|         | Output compare register D00                    | OCRD00       | 32             | H'FFFFF590 | 32             |
|         | Output compare register D01                    | OCRD01       | 32             | H'FFFFF594 | 32             |
|         | Output compare register D02                    | OCRD02       | 32             | H'FFFFF598 | 32             |
|         | Output compare register D03                    | OCRD03       | 32             | H'FFFFF59C | 32             |
|         | General register D00                           | GRD00        | 32             | H'FFFFF5A0 | 32             |
|         | General register D01                           | GRD01        | 32             | H'FFFFF5A4 | 32             |
|         | General register D02                           | GRD02        | 32             | H'FFFFF5A8 | 32             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | General register D03               | GRD03        | 32             | H'FFFFF5AC | 32             |
|         | Timer down counter D00             | DCNTD00      | 32             | H'FFFFF5B0 | 32             |
|         | Timer down counter D01             | DCNTD01      | 32             | H'FFFFF5B4 | 32             |
|         | Timer down counter D02             | DCNTD02      | 32             | H'FFFFF5B8 | 32             |
|         | Timer down counter D03             | DCNTD03      | 32             | H'FFFFF5BC | 32             |
|         | Timer I/O control register 1D1     | TIOR1D1      | 16             | H'FFFFF5C0 | 8, 16          |
|         | Timer I/O control register 2D1     | TIOR2D1      | 16             | H'FFFFF5C2 | 8, 16          |
|         | Down counter starting register D1  | DSTRD1       | 8              | H'FFFFF5C5 | 8              |
|         | Down counter status register D1    | DSRD1        | 8              | H'FFFFF5C7 | 8              |
|         | Down counter control register D1   | DCRD1        | 16             | H'FFFFF5C8 | 8, 16          |
|         | Timer status register D1           | TSRD1        | 16             | H'FFFFF5CC | 8, 16          |
|         | Timer interrupt enable register D1 | TIERD1       | 16             | H'FFFFF5CE | 8, 16          |
|         | Output compare register D10        | OCRD10       | 32             | H'FFFFF5D0 | 32             |
|         | Output compare register D11        | OCRD11       | 32             | H'FFFFF5D4 | 32             |
|         | Output compare register D12        | OCRD12       | 32             | H'FFFFF5D8 | 32             |
|         | Output compare register D13        | OCRD13       | 32             | H'FFFFF5DC | 32             |
|         | General register D10               | GRD10        | 32             | H'FFFFF5E0 | 32             |
|         | General register D11               | GRD11        | 32             | H'FFFFF5E4 | 32             |
|         | General register D12               | GRD12        | 32             | H'FFFFF5E8 | 32             |
|         | General register D13               | GRD13        | 32             | H'FFFFF5EC | 32             |
|         | Timer down counter D10             | DCNTD10      | 32             | H'FFFFF5F0 | 32             |
|         | Timer down counter D11             | DCNTD11      | 32             | H'FFFFF5F4 | 32             |
|         | Timer down counter D12             | DCNTD12      | 32             | H'FFFFF5F8 | 32             |
|         | Timer down counter D13             | DCNTD13      | 32             | H'FFFFF5FC | 32             |
|         | Timer I/O control register 1D2     | TIOR1D2      | 16             | H'FFFFF600 | 8, 16          |
| -       | Timer I/O control register 2D2     | TIOR2D2      | 16             | H'FFFFF602 | 8, 16          |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Down counter starting register D2  | DSTRD2       | 8              | H'FFFFF605 | 8              |
|         | Down counter status register D2    | DSRD2        | 8              | H'FFFFF607 | 8              |
|         | Down counter control register D2   | DCRD2        | 16             | H'FFFFF608 | 8, 16          |
|         | Timer status register D2           | TSRD2        | 16             | H'FFFFF60C | 8, 16          |
|         | Timer interrupt enable register D2 | TIERD2       | 16             | H'FFFFF60E | 8, 16          |
|         | Output compare register D20        | OCRD20       | 32             | H'FFFFF610 | 32             |
|         | Output compare register D21        | OCRD21       | 32             | H'FFFFF614 | 32             |
|         | Output compare register D22        | OCRD22       | 32             | H'FFFFF618 | 32             |
|         | Output compare register D23        | OCRD23       | 32             | H'FFFFF61C | 32             |
|         | General register D20               | GRD20        | 32             | H'FFFFF620 | 32             |
|         | General register D21               | GRD21        | 32             | H'FFFFF624 | 32             |
|         | General register D22               | GRD22        | 32             | H'FFFFF628 | 32             |
|         | General register D23               | GRD23        | 32             | H'FFFFF62C | 32             |
|         | Timer down counter D20             | DCNTD20      | 32             | H'FFFFF630 | 32             |
|         | Timer down counter D21             | DCNTD21      | 32             | H'FFFFF634 | 32             |
|         | Timer down counter D22             | DCNTD22      | 32             | H'FFFFF638 | 32             |
|         | Timer down counter D23             | DCNTD23      | 32             | H'FFFFF63C | 32             |
|         | Timer I/O control register 1D3     | TIOR1D3      | 16             | H'FFFFF640 | 8, 16          |
|         | Timer I/O control register 2D3     | TIOR2D3      | 16             | H'FFFFF642 | 8, 16          |
|         | Down counter starting register D3  | DSTRD3       | 8              | H'FFFFF645 | 8              |
|         | Down counter status register D3    | DSRD3        | 8              | H'FFFFF647 | 8              |
|         | Down counter control register D3   | DCRD3        | 16             | H'FFFFF648 | 8, 16          |
|         | Timer status register D3           | TSRD3        | 16             | H'FFFFF64C | 8, 16          |
|         | Timer interrupt enable register D3 | TIERD3       | 16             | H'FFFFF64E | 8, 16          |
|         | Output compare register D30        | OCRD30       | 32             | H'FFFFF650 | 32             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Output compare register D31        | OCRD31       | 32             | H'FFFFF654 | 32             |
|         | Output compare register D32        | OCRD32       | 32             | H'FFFFF658 | 32             |
|         | Output compare register D33        | OCRD33       | 32             | H'FFFFF65C | 32             |
|         | General register D30               | GRD30        | 32             | H'FFFFF660 | 32             |
|         | General register D31               | GRD31        | 32             | H'FFFFF664 | 32             |
|         | General register D32               | GRD32        | 32             | H'FFFFF668 | 32             |
|         | General register D33               | GRD33        | 32             | H'FFFFF66C | 32             |
|         | Timer down counter D30             | DCNTD30      | 32             | H'FFFFF670 | 32             |
|         | Timer down counter D31             | DCNTD31      | 32             | H'FFFFF674 | 32             |
|         | Timer down counter D32             | DCNTD32      | 32             | H'FFFFF678 | 32             |
|         | Timer down counter D33             | DCNTD33      | 32             | H'FFFFF67C | 32             |
|         | Timer start register E             | TSTRE        | 8              | H'FFFFF700 | 8              |
|         | Timer control register E0          | TCRE0        | 8              | H'FFFFF800 | 8              |
|         | Timer output control register E0   | TOCRE0       | 8              | H'FFFFF801 | 8              |
|         | Timer interrupt enable register E0 | TIERE0       | 8              | H'FFFFF802 | 8              |
|         | Reload control register E0         | RLDCRE0      | 8              | H'FFFFF803 | 8              |
|         | Timer status register E0           | TSRE0        | 8              | H'FFFFF804 | 8              |
|         | Prescaler register E0              | PSCRE0       | 8              | H'FFFFF808 | 8              |
|         | Subblock starting register E0      | SSTRE0       | 8              | H'FFFFF80C | 8              |
|         | Cycle-setting register E00         | CYLRE00      | 16             | H'FFFFF810 | 16             |
|         | Cycle-setting register E01         | CYLRE01      | 16             | H'FFFFF812 | 16             |
|         | Cycle-setting register E02         | CYLRE02      | 16             | H'FFFFF814 | 16             |
|         | Cycle-setting register E03         | CYLRE03      | 16             | H'FFFFF816 | 16             |
|         | Duty cycle setting register E00    | DTRE00       | 16             | H'FFFFF818 | 16             |
|         | Duty cycle setting register E01    | DTRE01       | 16             | H'FFFFF81A | 16             |
|         | Duty cycle setting register E02    | DTRE02       | 16             | H'FFFFF81C | 16             |
|         | Duty cycle setting register E03    | DTRE03       | 16             | H'FFFFF81E | 16             |
|         | Cycle reload register E00          | CRLDE00      | 16             | H'FFFFF820 | 16             |
|         | Cycle reload register E01          | CRLDE01      | 16             | H'FFFFF822 | 16             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Cycle reload register E02          | CRLDE02      | 16             | H'FFFFF824 | 16             |
|         | Cycle reload register E03          | CRLDE03      | 16             | H'FFFFF826 | 16             |
|         | Duty cycle reload register E00     | DRLDE00      | 16             | H'FFFFF828 | 16             |
|         | Duty cycle reload register E01     | DRLDE01      | 16             | H'FFFFF82A | 16             |
|         | Duty cycle reload register E02     | DRLDE02      | 16             | H'FFFFF82C | 16             |
|         | Duty cycle reload register E03     | DRLDE03      | 16             | H'FFFFF82E | 16             |
|         | Timer counter E00                  | TCNTE00      | 16             | H'FFFFF830 | 16             |
|         | Timer counter E01                  | TCNTE01      | 16             | H'FFFFF832 | 16             |
|         | Timer counter E02                  | TCNTE02      | 16             | H'FFFFF834 | 16             |
|         | Timer counter E03                  | TCNTE03      | 16             | H'FFFFF836 | 16             |
|         | Timer control register E1          | TCRE1        | 8              | H'FFFFF840 | 8              |
|         | Timer output control register E1   | TOCRE1       | 8              | H'FFFFF841 | 8              |
|         | Timer interrupt enable register E1 | TIERE1       | 8              | H'FFFFF842 | 8              |
|         | Reload control register E1         | RLDCRE1      | 8              | H'FFFFF843 | 8              |
|         | Timer status register E1           | TSRE1        | 8              | H'FFFFF844 | 8              |
|         | Prescaler register E1              | PSCRE1       | 8              | H'FFFFF848 | 8              |
|         | Subblock starting register E1      | SSTRE1       | 8              | H'FFFFF84C | 8              |
|         | Cycle-setting register E10         | CYLRE10      | 16             | H'FFFFF850 | 16             |
|         | Cycle-setting register E11         | CYLRE11      | 16             | H'FFFFF852 | 16             |
|         | Cycle-setting register E12         | CYLRE12      | 16             | H'FFFFF854 | 16             |
|         | Cycle-setting register E13         | CYLRE13      | 16             | H'FFFFF856 | 16             |
|         | Duty cycle setting register E10    | DTRE10       | 16             | H'FFFFF858 | 16             |
|         | Duty cycle setting register E11    | DTRE11       | 16             | H'FFFFF85A | 16             |
|         | Duty cycle setting register E12    | DTRE12       | 16             | H'FFFFF85C | 16             |
|         | Duty cycle setting register E13    | DTRE13       | 16             | H'FFFFF85E | 16             |
|         | Cycle reload register E10          | CRLDE10      | 16             | H'FFFFF860 | 16             |
|         | Cycle reload register E11          | CRLDE11      | 16             | H'FFFFF862 | 16             |
|         | Cycle reload register E12          | CRLDE12      | 16             | H'FFFFF864 | 16             |
|         | Cycle reload register E13          | CRLDE13      | 16             | H'FFFFF866 | 16             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Duty cycle reload register E10     | DRLDE10      | 16             | H'FFFFF868 | 16             |
|         | Duty cycle reload register E11     | DRLDE11      | 16             | H'FFFFF86A | 16             |
|         | Duty cycle reload register E12     | DRLDE12      | 16             | H'FFFFF86C | 16             |
|         | Duty cycle reload register E13     | DRLDE13      | 16             | H'FFFFF86E | 16             |
|         | Timer counter E10                  | TCNTE10      | 16             | H'FFFFF870 | 16             |
|         | Timer counter E11                  | TCNTE11      | 16             | H'FFFFF872 | 16             |
|         | Timer counter E12                  | TCNTE12      | 16             | H'FFFFF874 | 16             |
|         | Timer counter E13                  | TCNTE13      | 16             | H'FFFFF876 | 16             |
|         | Timer control register E2          | TCRE2        | 8              | H'FFFFF880 | 8              |
|         | Timer output control register E2   | TOCRE2       | 8              | H'FFFFF881 | 8              |
|         | Timer interrupt enable register E2 | TIERE2       | 8              | H'FFFFF882 | 8              |
|         | Reload control register E2         | RLDCRE2      | 8              | H'FFFFF883 | 8              |
|         | Timer status register E2           | TSRE2        | 8              | H'FFFFF884 | 8              |
|         | Prescaler register E2              | PSCRE2       | 8              | H'FFFFF888 | 8              |
|         | Subblock starting register E2      | SSTRE2       | 8              | H'FFFFF88C | 8              |
|         | Cycle-setting register E20         | CYLRE20      | 16             | H'FFFFF890 | 16             |
|         | Cycle-setting register E21         | CYLRE21      | 16             | H'FFFFF892 | 16             |
|         | Cycle-setting register E22         | CYLRE22      | 16             | H'FFFFF894 | 16             |
|         | Cycle-setting register E23         | CYLRE23      | 16             | H'FFFFF896 | 16             |
|         | Duty cycle setting register E20    | DTRE20       | 16             | H'FFFFF898 | 16             |
|         | Duty cycle setting register E21    | DTRE21       | 16             | H'FFFFF89A | 16             |
|         | Duty cycle setting register E22    | DTRE22       | 16             | H'FFFFF89C | 16             |
|         | Duty cycle setting register E23    | DTRE23       | 16             | H'FFFFF89E | 16             |
|         | Cycle-setting register E20         | CRLDE20      | 16             | H'FFFFF8A0 | 16             |
|         | Cycle-setting register E21         | CRLDE21      | 16             | H'FFFFF8A2 | 16             |
|         | Cycle-setting register E22         | CRLDE22      | 16             | H'FFFFF8A4 | 16             |
|         | Cycle-setting register E23         | CRLDE23      | 16             | H'FFFFF8A6 | 16             |
|         | Duty cycle reload register E20     | DRLDE20      | 16             | H'FFFFF8A8 | 16             |
|         | Duty cycle reload register E21     | DRLDE21      | 16             | H'FFFFF8AA | 16             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Duty cycle reload register E22     | DRLDE22      | 16             | H'FFFFF8AC | 16             |
|         | Duty cycle reload register E23     | DRLDE23      | 16             | H'FFFFF8AE | 16             |
|         | Timer counter E20                  | TCNTE20      | 16             | H'FFFFF8B0 | 16             |
|         | Timer counter E21                  | TCNTE21      | 16             | H'FFFFF8B2 | 16             |
|         | Timer counter E22                  | TCNTE22      | 16             | H'FFFFF8B4 | 16             |
|         | Timer counter E23                  | TCNTE23      | 16             | H'FFFFF8B6 | 16             |
|         | Timer control register E3          | TCRE3        | 8              | H'FFFFF8C0 | 8              |
|         | Timer output control register E3   | TOCRE3       | 8              | H'FFFFF8C1 | 8              |
|         | Timer interrupt enable register E3 | TIERE3       | 8              | H'FFFFF8C2 | 8              |
|         | Reload control register E3         | RLDCRE3      | 8              | H'FFFFF8C3 | 8              |
|         | Timer status register E3           | TSRE3        | 8              | H'FFFFF8C4 | 8              |
|         | Prescaler register E3              | PSCRE3       | 8              | H'FFFFF8C8 | 8              |
|         | Subblock starting register E3      | SSTRE3       | 8              | H'FFFFF8CC | 8              |
|         | Cycle-setting register E30         | CYLRE30      | 16             | H'FFFFF8D0 | 16             |
|         | Cycle-setting register E31         | CYLRE31      | 16             | H'FFFFF8D2 | 16             |
|         | Cycle-setting register E32         | CYLRE32      | 16             | H'FFFFF8D4 | 16             |
|         | Cycle-setting register E33         | CYLRE33      | 16             | H'FFFFF8D6 | 16             |
|         | Duty cycle setting register E30    | DTRE30       | 16             | H'FFFFF8D8 | 16             |
|         | Duty cycle setting register E31    | DTRE31       | 16             | H'FFFFF8DA | 16             |
|         | Duty cycle setting register E32    | DTRE32       | 16             | H'FFFFF8DC | 16             |
|         | Duty cycle setting register E33    | DTRE33       | 16             | H'FFFFF8DE | 16             |
|         | Cycle-setting register E30         | CRLDE30      | 16             | H'FFFFF8E0 | 16             |
|         | Cycle-setting register E31         | CRLDE31      | 16             | H'FFFFF8E2 | 16             |
|         | Cycle-setting register E32         | CRLDE32      | 16             | H'FFFFF8E4 | 16             |
|         | Cycle-setting register E33         | CRLDE33      | 16             | H'FFFFF8E6 | 16             |
|         | Duty cycle reload register E30     | DRLDE30      | 16             | H'FFFFF8E8 | 16             |
|         | Duty cycle reload register E31     | DRLDE31      | 16             | H'FFFFF8EA | 16             |
|         | Duty cycle reload register E32     | DRLDE32      | 16             | H'FFFFF8EC | 16             |
|         | Duty cycle reload register E33     | DRLDE33      | 16             | H'FFFFF8EE | 16             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Timer counter E30                  | TCNTE30      | 16             | H'FFFFF8F0 | 16             |
|         | Timer counter E31                  | TCNTE31      | 16             | H'FFFFF8F2 | 16             |
|         | Timer counter E32                  | TCNTE32      | 16             | H'FFFFF8F4 | 16             |
|         | Timer counter E33                  | TCNTE33      | 16             | H'FFFFF8F6 | 16             |
|         | Timer control register E4          | TCRE4        | 8              | H'FFFFF900 | 8              |
|         | Timer output control register E4   | TOCRE4       | 8              | H'FFFFF901 | 8              |
|         | Timer interrupt enable register E4 | TIERE4       | 8              | H'FFFFF902 | 8              |
|         | Reload control register E4         | RLDCRE4      | 8              | H'FFFFF903 | 8              |
|         | Timer status register E4           | TSRE4        | 8              | H'FFFFF904 | 8              |
|         | Prescaler register E4              | PSCRE4       | 8              | H'FFFFF908 | 8              |
|         | Subblock starting register E4      | SSTRE4       | 8              | H'FFFFF90C | 8              |
|         | Cycle-setting register E40         | CYLRE40      | 16             | H'FFFFF910 | 16             |
|         | Cycle-setting register E41         | CYLRE41      | 16             | H'FFFFF912 | 16             |
|         | Cycle-setting register E42         | CYLRE42      | 16             | H'FFFFF914 | 16             |
|         | Cycle-setting register E43         | CYLRE43      | 16             | H'FFFFF916 | 16             |
|         | Duty cycle setting register E40    | DTRE40       | 16             | H'FFFFF918 | 16             |
|         | Duty cycle setting register E41    | DTRE41       | 16             | H'FFFFF91A | 16             |
|         | Duty cycle setting register E42    | DTRE42       | 16             | H'FFFFF91C | 16             |
|         | Duty cycle setting register E43    | DTRE43       | 16             | H'FFFFF91E | 16             |
|         | Cycle-setting register E40         | CRLDE40      | 16             | H'FFFFF920 | 16             |
|         | Cycle-setting register E41         | CRLDE41      | 16             | H'FFFFF922 | 16             |
|         | Cycle-setting register E42         | CRLDE42      | 16             | H'FFFFF924 | 16             |
|         | Cycle-setting register E43         | CRLDE43      | 16             | H'FFFFF926 | 16             |
|         | Duty cycle reload register E40     | DRLDE40      | 16             | H'FFFFF928 | 16             |
|         | Duty cycle reload register E41     | DRLDE41      | 16             | H'FFFFF92A | 16             |
|         | Duty cycle reload register E42     | DRLDE42      | 16             | H'FFFFF92C | 16             |
|         | Duty cycle reload register E43     | DRLDE43      | 16             | H'FFFFF92E | 16             |
|         | Timer counter E40                  | TCNTE40      | 16             | H'FFFFF930 | 16             |
|         | Timer counter E41                  | TCNTE41      | 16             | H'FFFFF932 | 16             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Timer counter E42                  | TCNTE42      | 16             | H'FFFFF934 | 16             |
|         | Timer counter E43                  | TCNTE43      | 16             | H'FFFFF936 | 16             |
|         | Timer control register E5          | TCRE5        | 8              | H'FFFFF940 | 8              |
|         | Timer output control register E5   | TOCRE5       | 8              | H'FFFFF941 | 8              |
|         | Timer interrupt enable register E5 | TIERE5       | 8              | H'FFFFF942 | 8              |
|         | Reload control register E5         | RLDCRE5      | 8              | H'FFFFF943 | 8              |
|         | Timer status register E5           | TSRE5        | 8              | H'FFFFF944 | 8              |
|         | Prescaler register E5              | PSCRE5       | 8              | H'FFFFF948 | 8              |
|         | Subblock starting register E5      | SSTRE5       | 8              | H'FFFFF94C | 8              |
|         | Cycle-setting register E50         | CYLRE50      | 16             | H'FFFFF950 | 16             |
|         | Cycle-setting register E51         | CYLRE51      | 16             | H'FFFFF952 | 16             |
|         | Cycle-setting register E52         | CYLRE52      | 16             | H'FFFFF954 | 16             |
|         | Cycle-setting register E53         | CYLRE53      | 16             | H'FFFFF956 | 16             |
|         | Duty cycle setting register E50    | DTRE50       | 16             | H'FFFFF958 | 16             |
|         | Duty cycle setting register E51    | DTRE51       | 16             | H'FFFFF95A | 16             |
|         | Duty cycle setting register E52    | DTRE52       | 16             | H'FFFFF95C | 16             |
|         | Duty cycle setting register E53    | DTRE53       | 16             | H'FFFFF95E | 16             |
|         | Cycle-setting register E50         | CRLDE50      | 16             | H'FFFFF960 | 16             |
|         | Cycle-setting register E51         | CRLDE51      | 16             | H'FFFFF962 | 16             |
|         | Cycle-setting register E52         | CRLDE52      | 16             | H'FFFFF964 | 16             |
|         | Cycle-setting register E53         | CRLDE53      | 16             | H'FFFFF966 | 16             |
|         | Duty cycle reload register E50     | DRLDE50      | 16             | H'FFFFF968 | 16             |
|         | Duty cycle reload register E51     | DRLDE51      | 16             | H'FFFFF96A | 16             |
|         | Duty cycle reload register E52     | DRLDE52      | 16             | H'FFFFF96C | 16             |
|         | Duty cycle reload register E53     | DRLDE53      | 16             | H'FFFFF96E | 16             |
|         | Timer counter E50                  | TCNTE50      | 16             | H'FFFFF970 | 16             |
|         | Timer counter E51                  | TCNTE51      | 16             | H'FFFFF972 | 16             |
|         | Timer counter E52                  | TCNTE52      | 16             | H'FFFFF974 | 16             |
|         | Timer counter E53                  | TCNTE53      | 16             | H'FFFFF976 | 16             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Timer start register F             | TSTRF        | 32             | H'FFFFFA00 | 8, 16, 32      |
|         | Noise canceller control register F | NCCRF        | 32             | H'FFFFFA04 | 8, 16, 32      |
|         | Noise canceler counter FA0         | NCNTFA0      | 8              | H'FFFFFA10 | 8              |
|         | Noise cancel register FA0          | NCRFA0       | 8              | H'FFFFFA11 | 8              |
|         | Noise canceler counter FA1         | NCNTFA1      | 8              | H'FFFFFA12 | 8              |
|         | Noise cancel register FA1          | NCRFA1       | 8              | H'FFFFFA13 | 8              |
|         | Noise canceler counter FA2         | NCNTFA2      | 8              | H'FFFFFA14 | 8              |
|         | Noise cancel register FA2          | NCRFA2       | 8              | H'FFFFFA15 | 8              |
|         | Noise canceler counter FA3         | NCNTFA3      | 8              | H'FFFFFA16 | 8              |
|         | Noise cancel register FA3          | NCRFA3       | 8              | H'FFFFFA17 | 8              |
|         | Noise canceler counter FA4         | NCNTFA4      | 8              | H'FFFFFA18 | 8              |
|         | Noise cancel register FA4          | NCRFA4       | 8              | H'FFFFFA19 | 8              |
|         | Noise canceler counter FA5         | NCNTFA5      | 8              | H'FFFFFA1A | 8              |
|         | Noise cancel register FA5          | NCRFA5       | 8              | H'FFFFFA1B | 8              |
|         | Noise canceler counter FA6         | NCNTFA6      | 8              | H'FFFFFA1C | 8              |
|         | Noise cancel register FA6          | NCRFA6       | 8              | H'FFFFFA1D | 8              |
|         | Noise canceler counter FA7         | NCNTFA7      | 8              | H'FFFFFA1E | 8              |
|         | Noise cancel register FA7          | NCRFA7       | 8              | H'FFFFFA1F | 8              |
|         | Noise canceler counter FA8         | NCNTFA8      | 8              | H'FFFFFA20 | 8              |
|         | Noise cancel register FA8          | NCRFA8       | 8              | H'FFFFFA21 | 8              |
|         | Noise canceler counter FA9         | NCNTFA9      | 8              | H'FFFFFA22 | 8              |
|         | Noise cancel register FA9          | NCRFA9       | 8              | H'FFFFFA23 | 8              |
|         | Noise canceler counter FA10        | NCNTFA10     | 8              | H'FFFFFA24 | 8              |
|         | Noise cancel register FA10         | NCRFA10      | 8              | H'FFFFFA25 | 8              |
|         | Noise canceler counter FA11        | NCNTFA11     | 8              | H'FFFFFA26 | 8              |
|         | Noise cancel register FA11         | NCRFA11      | 8              | H'FFFFFA27 | 8              |
|         | Noise canceler counter FA12        | NCNTFA12     | 8              | H'FFFFFA28 | 8              |
|         | Noise cancel register FA12         | NCRFA12      | 8              | H'FFFFFA29 | 8              |
|         | Noise canceler counter FA13        | NCNTFA13     | 8              | H'FFFFFA2A | 8              |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Noise cancel register FA13         | NCRFA13      | 8              | H'FFFFFA2B | 8              |
|         | Noise canceler counter FA14        | NCNTFA14     | 8              | H'FFFFFA2C | 8              |
|         | Noise cancel register FA14         | NCRFA14      | 8              | H'FFFFFA2D | 8              |
|         | Noise canceler counter FA15        | NCNTFA15     | 8              | H'FFFFFA2E | 8              |
|         | Noise cancel register FA15         | NCRFA15      | 8              | H'FFFFFA2F | 8              |
|         | Noise canceler counter FA16        | NCNTFA16     | 8              | H'FFFFFA30 | 8              |
|         | Noise cancel register FA16         | NCRFA16      | 8              | H'FFFFFA31 | 8              |
|         | Noise canceler counter FA17        | NCNTFA17     | 8              | H'FFFFFA32 | 8              |
|         | Noise cancel register FA17         | NCRFA17      | 8              | H'FFFFFA33 | 8              |
|         | Noise canceler counter FA18        | NCNTFA18     | 8              | H'FFFFFA34 | 8              |
|         | Noise cancel register FA18         | NCRFA18      | 8              | H'FFFFFA35 | 8              |
|         | Noise canceler counter FA19        | NCNTFA19     | 8              | H'FFFFFA36 | 8              |
|         | Noise cancel register FA19         | NCRFA19      | 8              | H'FFFFFA37 | 8              |
|         | Noise canceler counter FB0         | NCNTFB0      | 8              | H'FFFFFA50 | 8              |
|         | Noise cancel register FB0          | NCRFB0       | 8              | H'FFFFFA51 | 8              |
|         | Noise canceler counter FB1         | NCNTFB1      | 8              | H'FFFFFA52 | 8              |
|         | Noise cancel register FB1          | NCRFB1       | 8              | H'FFFFFA53 | 8              |
|         | Noise canceler counter FB2         | NCNTFB2      | 8              | H'FFFFFA54 | 8              |
|         | Noise cancel register FB2          | NCRFB2       | 8              | H'FFFFFA55 | 8              |
|         | Timer control register F0          | TCRF0        | 8              | H'FFFFFA80 | 8              |
|         | Timer interrupt enable register F0 | TIERF0       | 8              | H'FFFFFA81 | 8              |
|         | Timer status register F0           | TSRF0        | 8              | H'FFFFFA83 | 8              |
|         | Timer counter AF0                  | ECNTAF0      | 32             | H'FFFFFA84 | 32             |
|         | Event counter BF0                  | ECNTBF0      | 16             | H'FFFFFA88 | 16             |
|         | General register BF0               | GRBF0        | 16             | H'FFFFFA8A | 16             |
|         | Time counter CF0                   | ECNTCF0      | 32             | H'FFFFFA8C | 32             |
|         | General register AF0               | GRAF0        | 32             | H'FFFFFA90 | 32             |
|         | Capture output register F0         | CDRF0        | 32             | H'FFFFFA94 | 32             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | General register CF0               | GRCF0        | 32             | H'FFFFFA98 | 32             |
|         | Timer control register F1          | TCRF1        | 8              | H'FFFFFAA0 | 8              |
|         | Timer interrupt enable register F1 | TIERF1       | 8              | H'FFFFFAA1 | 8              |
|         | Timer status register F1           | TSRF1        | 8              | H'FFFFFAA3 | 8              |
|         | Timer counter AF1                  | ECNTAF1      | 32             | H'FFFFFAA4 | 32             |
|         | Event counter BF1                  | ECNTBF1      | 16             | H'FFFFFAA8 | 16             |
|         | General register BF1               | GRBF1        | 16             | H'FFFFFAAA | 16             |
|         | Time counter CF1                   | ECNTCF1      | 32             | H'FFFFFAAC | 32             |
|         | General register AF1               | GRAF1        | 32             | H'FFFFFAB0 | 32             |
|         | Capture output register F1         | CDRF1        | 32             | H'FFFFFAB4 | 32             |
|         | General register CF1               | GRCF1        | 32             | H'FFFFFAB8 | 32             |
|         | Timer control register F2          | TCRF2        | 8              | H'FFFFFAC0 | 8              |
|         | Timer interrupt enable register F2 | TIERF2       | 8              | H'FFFFFAC1 | 8              |
|         | Timer status register F2           | TSRF2        | 8              | H'FFFFFAC3 | 8              |
|         | Timer counter AF2                  | ECNTAF2      | 32             | H'FFFFFAC4 | 32             |
|         | Event counter BF2                  | ECNTBF2      | 16             | H'FFFFFAC8 | 16             |
|         | General register BF2               | GRBF2        | 16             | H'FFFFFACA | 16             |
|         | Time counter CF2                   | ECNTCF2      | 32             | H'FFFFFACC | 32             |
|         | General register AF2               | GRAF2        | 32             | H'FFFFFAD0 | 32             |
|         | Capture output register F2         | CDRF2        | 32             | H'FFFFFAD4 | 32             |
|         | General register CF2               | GRCF2        | 32             | H'FFFFFAD8 | 32             |
|         | Timer control register F3          | TCRF3        | 8              | H'FFFFFAE0 | 8              |
|         | Timer interrupt enable register F3 | TIERF3       | 8              | H'FFFFFAE1 | 8              |
|         | Timer status register F3           | TSRF3        | 8              | H'FFFFFAE3 | 8              |
|         | Timer counter AF3                  | ECNTAF3      | 32             | H'FFFFFAE4 | 32             |
|         | Event counter BF3                  | ECNTBF3      | 16             | H'FFFFFAE8 | 16             |
|         | General register BF3               | GRBF3        | 16             | H'FFFFFAEA | 16             |
|         | Time counter CF3                   | ECNTCF3      | 32             | H'FFFFFAEC | 32             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | General register AF3               | GRAF3        | 32             | H'FFFFFAF0 | 32             |
|         | Capture output register F3         | CDRF3        | 32             | H'FFFFFAF4 | 32             |
|         | General register CF3               | GRCF3        | 32             | H'FFFFFAF8 | 32             |
|         | Timer control register F4          | TCRF4        | 8              | H'FFFFFB00 | 8              |
|         | Timer interrupt enable register F4 | TIERF4       | 8              | H'FFFFFB01 | 8              |
|         | Timer status register F4           | TSRF4        | 8              | H'FFFFFB03 | 8              |
|         | Timer counter AF4                  | ECNTAF4      | 32             | H'FFFFFB04 | 32             |
|         | Event counter BF4                  | ECNTBF4      | 16             | H'FFFFFB08 | 16             |
|         | General register BF4               | GRBF4        | 16             | H'FFFFFB0A | 16             |
|         | Time counter CF4                   | ECNTCF4      | 32             | H'FFFFFB0C | 32             |
|         | General register AF4               | GRAF4        | 32             | H'FFFFB10  | 32             |
|         | Capture output register F4         | CDRF4        | 32             | H'FFFFB14  | 32             |
|         | General register CF4               | GRCF4        | 32             | H'FFFFB18  | 32             |
|         | Timer control register F5          | TCRF5        | 8              | H'FFFFB20  | 8              |
|         | Timer interrupt enable register F5 | TIERF5       | 8              | H'FFFFFB21 | 8              |
|         | Timer status register F5           | TSRF5        | 8              | H'FFFFFB23 | 8              |
|         | Timer counter AF5                  | ECNTAF5      | 32             | H'FFFFFB24 | 32             |
|         | Event counter BF5                  | ECNTBF5      | 16             | H'FFFFFB28 | 16             |
|         | General register BF5               | GRBF5        | 16             | H'FFFFFB2A | 16             |
|         | Time counter CF5                   | ECNTCF5      | 32             | H'FFFFFB2C | 32             |
|         | General register AF5               | GRAF5        | 32             | H'FFFFFB30 | 32             |
|         | Capture output register F5         | CDRF5        | 32             | H'FFFFFB34 | 32             |
|         | General register CF5               | GRCF5        | 32             | H'FFFFFB38 | 32             |
|         | Timer control register F6          | TCRF6        | 8              | H'FFFFFB40 | 8              |
|         | Timer interrupt enable register F6 | TIERF6       | 8              | H'FFFFFB41 | 8              |
|         | Timer status register F6           | TSRF6        | 8              | H'FFFFFB43 | 8              |
|         | Timer counter AF6                  | ECNTAF6      | 32             | H'FFFFFB44 | 32             |
|         | Event counter BF6                  | ECNTBF6      | 16             | H'FFFFFB48 | 16             |

| Module  | Register Name                      | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | General register BF6               | GRBF6        | 16             | H'FFFFFB4A | 16             |
|         | Time counter CF6                   | ECNTCF6      | 32             | H'FFFFFB4C | 32             |
|         | General register AF6               | GRAF6        | 32             | H'FFFFFB50 | 32             |
|         | Capture output register F6         | CDRF6        | 32             | H'FFFFFB54 | 32             |
|         | General register CF6               | GRCF6        | 32             | H'FFFFFB58 | 32             |
|         | Timer control register F7          | TCRF7        | 8              | H'FFFFFB60 | 8              |
|         | Timer interrupt enable register F7 | TIERF7       | 8              | H'FFFFFB61 | 8              |
|         | Timer status register F7           | TSRF7        | 8              | H'FFFFFB63 | 8              |
|         | Timer counter AF7                  | ECNTAF7      | 32             | H'FFFFFB64 | 32             |
|         | Event counter BF7                  | ECNTBF7      | 16             | H'FFFFFB68 | 16             |
|         | General register BF7               | GRBF7        | 16             | H'FFFFFB6A | 16             |
|         | Time counter CF7                   | ECNTCF7      | 32             | H'FFFFFB6C | 32             |
|         | General register AF7               | GRAF7        | 32             | H'FFFFFB70 | 32             |
|         | Capture output register F7         | CDRF7        | 32             | H'FFFFFB74 | 32             |
|         | General register CF7               | GRCF7        | 32             | H'FFFFFB78 | 32             |
|         | Timer control register F8          | TCRF8        | 8              | H'FFFFFB80 | 8              |
|         | Timer interrupt enable register F8 | TIERF8       | 8              | H'FFFFFB81 | 8              |
|         | Timer status register F8           | TSRF8        | 8              | H'FFFFFB83 | 8              |
|         | Timer counter AF8                  | ECNTAF8      | 32             | H'FFFFFB84 | 32             |
|         | Event counter BF8                  | ECNTBF8      | 16             | H'FFFFFB88 | 16             |
|         | General register BF8               | GRBF8        | 16             | H'FFFFFB8A | 16             |
|         | Time counter CF8                   | ECNTCF8      | 32             | H'FFFFFB8C | 32             |
|         | General register AF8               | GRAF8        | 32             | H'FFFFFB90 | 32             |
|         | Capture output register F8         | CDRF8        | 32             | H'FFFFFB94 | 32             |
|         | General register CF8               | GRCF8        | 32             | H'FFFFFB98 | 32             |
|         | Timer control register F9          | TCRF9        | 8              | H'FFFFBA0  | 8              |
|         | Timer interrupt enable register F9 | TIERF9       | 8              | H'FFFFFBA1 | 8              |
|         | Timer status register F9           | TSRF9        | 8              | H'FFFFFBA3 | 8              |

| Module  | Register Name                       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|-------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Timer counter AF9                   | ECNTAF9      | 32             | H'FFFFFBA4 | 32             |
|         | Event counter BF9                   | ECNTBF9      | 16             | H'FFFFFBA8 | 16             |
|         | General register BF9                | GRBF9        | 16             | H'FFFFBAA  | 16             |
|         | Time counter CF9                    | ECNTCF9      | 32             | H'FFFFBAC  | 32             |
|         | General register AF9                | GRAF9        | 32             | H'FFFFBB0  | 32             |
|         | Capture output register F9          | CDRF9        | 32             | H'FFFFBB4  | 32             |
|         | General register CF9                | GRCF9        | 32             | H'FFFFFBB8 | 32             |
|         | Timer control register F10          | TCRF10       | 8              | H'FFFFBC0  | 8              |
|         | Timer interrupt enable register F10 | TIERF10      | 8              | H'FFFFFBC1 | 8              |
|         | Timer status register F10           | TSRF10       | 8              | H'FFFFBC3  | 8              |
|         | Timer counter AF10                  | ECNTAF10     | 32             | H'FFFFBC4  | 32             |
|         | Event counter BF10                  | ECNTBF10     | 16             | H'FFFFBC8  | 16             |
|         | General register BF10               | GRBF10       | 16             | H'FFFFBCA  | 16             |
|         | Time counter CF10                   | ECNTCF10     | 32             | H'FFFFFBCC | 32             |
|         | General register AF10               | GRAF10       | 32             | H'FFFFBD0  | 32             |
|         | Capture output register F10         | CDRF10       | 32             | H'FFFFBD4  | 32             |
|         | General register CF10               | GRCF10       | 32             | H'FFFFBD8  | 32             |
|         | Timer control register F11          | TCRF11       | 8              | H'FFFFFBE0 | 8              |
|         | Timer interrupt enable register F11 | TIERF11      | 8              | H'FFFFFBE1 | 8              |
|         | Timer status register F11           | TSRF11       | 8              | H'FFFFFBE3 | 8              |
|         | Timer counter AF11                  | ECNTAF11     | 32             | H'FFFFFBE4 | 32             |
|         | Event counter BF11                  | ECNTBF11     | 16             | H'FFFFBE8  | 16             |
|         | General register BF11               | GRBF11       | 16             | H'FFFFBEA  | 16             |
|         | Time counter CF11                   | ECNTCF11     | 32             | H'FFFFFBEC | 32             |
|         | General register AF11               | GRAF11       | 32             | H'FFFFFBF0 | 32             |
|         | Capture output register F11         | CDRF11       | 32             | H'FFFFFBF4 | 32             |
|         | General register CF11               | GRCF11       | 32             | H'FFFFFBF8 | 32             |
|         | Timer control register F12          | TCRF12       | 8              | H'FFFFFC00 | 8              |

| Module  | Register Name                       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|-------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Timer interrupt enable register F12 | TIERF12      | 8              | H'FFFFFC01 | 8              |
|         | Timer status register F12           | TSRF12       | 8              | H'FFFFC03  | 8              |
|         | Timer counter AF12                  | ECNTAF12     | 32             | H'FFFFC04  | 32             |
|         | Event counter BF12                  | ECNTBF12     | 16             | H'FFFFFC08 | 16             |
|         | General register BF12               | GRBF12       | 16             | H'FFFFFC0A | 16             |
|         | Time counter CF12                   | ECNTCF12     | 32             | H'FFFFFC0C | 32             |
|         | General register AF12               | GRAF12       | 32             | H'FFFFFC10 | 32             |
|         | Capture output register F12         | CDRF12       | 32             | H'FFFFFC14 | 32             |
|         | General register CF12               | GRCF12       | 32             | H'FFFFFC18 | 32             |
|         | General register DF12               | GRDF12       | 32             | H'FFFFFC1C | 32             |
|         | Timer control register F13          | TCRF13       | 8              | H'FFFFFC20 | 8              |
|         | Timer interrupt enable register F13 | TIERF13      | 8              | H'FFFFFC21 | 8              |
|         | Timer status register F13           | TSRF13       | 8              | H'FFFFFC23 | 8              |
|         | Timer counter AF13                  | ECNTAF13     | 32             | H'FFFFFC24 | 32             |
|         | Event counter BF13                  | ECNTBF13     | 16             | H'FFFFFC28 | 16             |
|         | General register BF13               | GRBF13       | 16             | H'FFFFFC2A | 16             |
|         | Time counter CF13                   | ECNTCF13     | 32             | H'FFFFFC2C | 32             |
|         | General register AF13               | GRAF13       | 32             | H'FFFFFC30 | 32             |
|         | Capture output register F13         | CDRF13       | 32             | H'FFFFFC34 | 32             |
|         | General register CF13               | GRCF13       | 32             | H'FFFFFC38 | 32             |
|         | General register DF13               | GRDF13       | 32             | H'FFFFFC3C | 32             |
|         | Timer control register F14          | TCRF14       | 8              | H'FFFFFC40 | 8              |
|         | Timer interrupt enable register F14 | TIERF14      | 8              | H'FFFFFC41 | 8              |
|         | Timer status register F14           | TSRF14       | 8              | H'FFFFFC43 | 8              |
|         | Timer counter AF14                  | ECNTAF14     | 32             | H'FFFFFC44 | 32             |
|         | Event counter BF14                  | ECNTBF14     | 16             | H'FFFFFC48 | 16             |
|         | General register BF14               | GRBF14       | 16             | H'FFFFFC4A | 16             |
|         | Time counter CF14                   | ECNTCF14     | 32             | H'FFFFFC4C | 32             |

| Module  | Register Name                       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|-------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | General register AF14               | GRAF14       | 32             | H'FFFFFC50 | 32             |
|         | Capture output register F14         | CDRF14       | 32             | H'FFFFFC54 | 32             |
|         | General register CF14               | GRCF14       | 32             | H'FFFFFC58 | 32             |
|         | General register DF14               | GRDF14       | 32             | H'FFFFFC5C | 32             |
|         | Timer control register F15          | TCRF15       | 8              | H'FFFFFC60 | 8              |
|         | Timer interrupt enable register F15 | TIERF15      | 8              | H'FFFFFC61 | 8              |
|         | Timer status register F15           | TSRF15       | 8              | H'FFFFFC63 | 8              |
|         | Timer counter AF15                  | ECNTAF15     | 32             | H'FFFFFC64 | 32             |
|         | Event counter BF15                  | ECNTBF15     | 16             | H'FFFFFC68 | 16             |
|         | General register BF15               | GRBF15       | 16             | H'FFFFFC6A | 16             |
|         | Time counter CF15                   | ECNTCF15     | 32             | H'FFFFFC6C | 32             |
|         | General register AF15               | GRAF15       | 32             | H'FFFFFC70 | 32             |
|         | Capture output register F15         | CDRF15       | 32             | H'FFFFFC74 | 32             |
|         | General register CF15               | GRCF15       | 32             | H'FFFFFC78 | 32             |
|         | General register DF15               | GRDF15       | 32             | H'FFFFFC7C | 32             |
|         | Timer control register F16          | TCRF16       | 8              | H'FFFFFC80 | 8              |
|         | Timer interrupt enable register F16 | TIERF16      | 8              | H'FFFFFC81 | 8              |
|         | Timer status register F16           | TSRF16       | 8              | H'FFFFFC83 | 8              |
|         | Timer counter AF16                  | ECNTAF16     | 32             | H'FFFFFC84 | 32             |
|         | Event counter BF16                  | ECNTBF16     | 16             | H'FFFFFC88 | 16             |
|         | General register BF16               | GRBF16       | 16             | H'FFFFFC8A | 16             |
|         | Time counter CF16                   | ECNTCF16     | 32             | H'FFFFFC8C | 32             |
|         | General register AF16               | GRAF16       | 32             | H'FFFFFC90 | 32             |
|         | Capture output register F16         | CDRF16       | 32             | H'FFFFFC94 | 32             |
|         | General register CF16               | GRCF16       | 32             | H'FFFFFC98 | 32             |
|         | Timer control register F17          | TCRF17       | 8              | H'FFFFFCA0 | 8              |
|         | Timer interrupt enable register F17 | TIERF17      | 8              | H'FFFFFCA1 | 8              |
|         | Timer status register F17           | TSRF17       | 8              | H'FFFFFCA3 | 8              |

| Module  | Register Name                       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|-------------------------------------|--------------|----------------|------------|----------------|
| ATU-III | Timer counter AF17                  | ECNTAF17     | 32             | H'FFFFFCA4 | 32             |
|         | Event counter BF17                  | ECNTBF17     | 16             | H'FFFFFCA8 | 16             |
|         | General register BF17               | GRBF17       | 16             | H'FFFFFCAA | 16             |
|         | Time counter CF17                   | ECNTCF17     | 32             | H'FFFFFCAC | 32             |
|         | General register AF17               | GRAF17       | 32             | H'FFFFCB0  | 32             |
|         | Capture output register F17         | CDRF17       | 32             | H'FFFFCB4  | 32             |
|         | General register CF17               | GRCF17       | 32             | H'FFFFCB8  | 32             |
|         | Timer control register F18          | TCRF18       | 8              | H'FFFFCC0  | 8              |
|         | Timer interrupt enable register F18 | TIERF18      | 8              | H'FFFFFCC1 | 8              |
|         | Timer status register F18           | TSRF18       | 8              | H'FFFFFCC3 | 8              |
|         | Timer counter AF18                  | ECNTAF18     | 32             | H'FFFFCC4  | 32             |
|         | Event counter BF18                  | ECNTBF18     | 16             | H'FFFFFCC8 | 16             |
|         | General register BF18               | GRBF18       | 16             | H'FFFFFCCA | 16             |
|         | Time counter CF18                   | ECNTCF18     | 32             | H'FFFFFCCC | 32             |
|         | General register AF18               | GRAF18       | 32             | H'FFFFFCD0 | 32             |
|         | Capture output register F18         | CDRF18       | 32             | H'FFFFFCD4 | 32             |
|         | General register CF18               | GRCF18       | 32             | H'FFFFFCD8 | 32             |
|         | Timer control register F19          | TCRF19       | 8              | H'FFFFFCE0 | 8              |
|         | Timer interrupt enable register F19 | TIERF19      | 8              | H'FFFFFCE1 | 8              |
|         | Timer status register F19           | TSRF19       | 8              | H'FFFFFCE3 | 8              |
|         | Timer counter AF19                  | ECNTAF19     | 32             | H'FFFFFCE4 | 32             |
|         | Event counter BF19                  | ECNTBF19     | 16             | H'FFFFFCE8 | 16             |
|         | General register BF19               | GRBF19       | 16             | H'FFFFCEA  | 16             |
|         | Time counter CF19                   | ECNTCF19     | 32             | H'FFFFFCEC | 32             |
|         | General register AF19               | GRAF19       | 32             | H'FFFFFCF0 | 32             |
|         | Capture output register F19         | CDRF19       | 32             | H'FFFFFCF4 | 32             |
|         | General register CF19               | GRCF19       | 32             | H'FFFFFCF8 | 32             |
|         | Timer start register G              | TSTRG        | 8              | H'FFFFFE01 | 8              |

| Module  | Register Name             | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|---------------------------|--------------|----------------|------------|----------------|
| ATU-III | Timer control register G0 | TCRG0        | 8              | H'FFFFFE80 | 8              |
|         | Timer status register G0  | TSRG0        | 8              | H'FFFFFE81 | 8              |
|         | Timer counter G0          | TCNTG0       | 16             | H'FFFFFE84 | 16             |
|         | Compare match register G0 | OCRG0        | 16             | H'FFFFFE86 | 16             |
|         | Timer control register G1 | TCRG1        | 8              | H'FFFFFE90 | 8              |
|         | Timer status register G1  | TSRG1        | 8              | H'FFFFFE91 | 8              |
|         | Timer counter G1          | TCNTG1       | 16             | H'FFFFFE94 | 16             |
|         | Compare match register G1 | OCRG1        | 16             | H'FFFFFE96 | 16             |
|         | Timer control register G2 | TCRG2        | 8              | H'FFFFFEA0 | 8              |
|         | Timer status register G2  | TSRG2        | 8              | H'FFFFFEA1 | 8              |
|         | Timer counter G2          | TCNTG2       | 16             | H'FFFFFEA4 | 16             |
|         | Compare match register G2 | OCRG2        | 16             | H'FFFFFEA6 | 16             |
|         | Timer control register G3 | TCRG3        | 8              | H'FFFFFEB0 | 8              |
|         | Timer status register G3  | TSRG3        | 8              | H'FFFFFEB1 | 8              |
|         | Timer counter G3          | TCNTG3       | 16             | H'FFFFFEB4 | 16             |
|         | Compare match register G3 | OCRG3        | 16             | H'FFFFFEB6 | 16             |
|         | Timer control register G4 | TCRG4        | 8              | H'FFFFFEC0 | 8              |
|         | Timer status register G4  | TSRG4        | 8              | H'FFFFFEC1 | 8              |
|         | Timer counter G4          | TCNTG4       | 16             | H'FFFFFEC4 | 16             |
|         | Compare match register G4 | OCRG4        | 16             | H'FFFFFEC6 | 16             |
|         | Timer control register G5 | TCRG5        | 8              | H'FFFFFED0 | 8              |
|         | Timer status register G5  | TSRG5        | 8              | H'FFFFFED1 | 8              |
|         | Timer counter G5          | TCNTG5       | 16             | H'FFFFFED4 | 16             |
|         | Compare match register G5 | OCRG5        | 16             | H'FFFFFED6 | 16             |
|         | Timer control register H  | TCRH         | 8              | H'FFFFFF40 | 8              |
|         | Timer status register H   | TSRH         | 8              | H'FFFFFF41 | 8              |
|         | Timer counter 1H          | TCNT1H       | 16             | H'FFFFFF44 | 16             |
|         | Compare match register 1H | OCR1H        | 16             | H'FFFFFF46 | 16             |
|         | Timer counter 2H          | TCNT2H       | 32             | H'FFFFFF48 | 32             |
|         | Timer start register J    | TSTRJ        | 8              | H'FFFFFF80 | 8              |

| Module  | Register Name                          | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------|--|--------------|----------------|------------|----------------|
| ATU-III | Timer control register J0              | TCRJ0        | 8              | H'FFFFFF90 | 8              |
|         | FIFO control register J0               | FCRJ0        | 8              | H'FFFFFF91 | 8              |
|         | Timer status register J0               | TSRJ0        | 8              | H'FFFFFF92 | 8              |
|         | Timer interrupt enable register J0     | TIERJ0       | 8              | H'FFFFFF94 | 8              |
|         | FIFO data count register J0            | FDNRJ0       | 8              | H'FFFFFF95 | 8              |
|         | Noise canceler counter J0              | NCNTJ0       | 8              | H'FFFFFF96 | 8              |
|         | Noise cancel register J0               | NCRJ0        | 8              | H'FFFFFF97 | 8              |
|         | Timer counter J0                       | TCNTJ0       | 16             | H'FFFFFF98 | 16             |
|         | Compare match register J0              | OCRJ0        | 16             | H'FFFFFF9A | 16             |
|         | FIFO register J0                       | FIFOJ0       | 16             | H'FFFFFF9C | 16             |
|         | Timer control register J1              | TCRJ1        | 8              | H'FFFFFA0  | 8              |
|         | FIFO control register J1               | FCRJ1        | 8              | H'FFFFFFA1 | 8              |
|         | Timer status register J1               | TSRJ1        | 8              | H'FFFFFFA2 | 8              |
|         | Timer interrupt enable register J1     | TIERJ1       | 8              | H'FFFFFA4  | 8              |
|         | FIFO data count register J1            | FDNRJ1       | 8              | H'FFFFFFA5 | 8              |
|         | Noise canceler counter J1              | NCNTJ1       | 8              | H'FFFFFFA6 | 8              |
|         | Noise cancel register J1               | NCRJ1        | 8              | H'FFFFFA7  | 8              |
|         | Timer counter J1                       | TCNTJ1       | 16             | H'FFFFFFA8 | 16             |
|         | Compare match register J1              | OCRJ1        | 16             | H'FFFFFAA  | 16             |
|         | FIFO register J1                       | FIFOJ1       | 16             | H'FFFFFAC  | 16             |
| WDT     | Watchdog timer control register        | WTCR         | 16             | H'FFFE0000 | 8, 16          |
|         | Watchdog timer counter                 | WTCNT        | 16             | H'FFFE0002 | 8, 16          |
|         | Watchdog timer status register         | WTSR         | 16             | H'FFFE0004 | 8, 16          |
|         | Watchdog reset control register        | WRCR         | 16             | H'FFFE0006 | 8, 16          |
| CMT     | Compare match timer start register     | CMSTR        | 16             | H'FFFEC000 | 16             |
|         | Compare match timer control register_0 | CMCR_0       | 8              | H'FFFEC010 | 8              |

| Module | Register Name                          | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|--|--------------|----------------|------------|----------------|
| CMT    | Compare match timer status register_0  | CMSR_0       | 8              | H'FFFEC011 | 8              |
|        | Compare match counter_0                | CMCNT_0      | 16             | H'FFFEC012 | 16             |
|        | Compare match constant register_0      | CMCOR_0      | 16             | H'FFFEC014 | 16             |
|        | Compare match timer control register_1 | CMCR_1       | 8              | H'FFFEC020 | 8              |
|        | Compare match timer status register_1  | CMSR_1       | 8              | H'FFFEC021 | 8              |
|        | Compare match counter_1                | CMCNT_1      | 16             | H'FFFEC022 | 16             |
|        | Compare match constant register_1      | CMCOR_1      | 16             | H'FFFEC024 | 16             |
| SCI    | Serial mode register                   | SCSMR1A      | 8              | H'FFFF8000 | 8              |
|        | Bit rate register                      | SCBRR1A      | 8              | H'FFFF8004 | 8              |
|        | Serial control register                | SCSCR1A      | 8              | H'FFFF8008 | 8              |
|        | Transmit data register                 | SCTDR1A      | 8              | H'FFFF800C | 8              |
|        | Serial status register                 | SCSSR1A      | 8              | H'FFFF8010 | 8              |
|        | Receive data register                  | SCRDR1A      | 8              | H'FFFF8014 | 8              |
|        | Serial mode register                   | SCSMR1B      | 8              | H'FFFF8800 | 8              |
|        | Bit rate register                      | SCBRR1B      | 8              | H'FFFF8804 | 8              |
|        | Serial control register                | SCSCR1B      | 8              | H'FFFF8808 | 8              |
|        | Transmit data register                 | SCTDR1B      | 8              | H'FFFF880C | 8              |
|        | Serial status register                 | SCSSR1B      | 8              | H'FFFF8810 | 8              |
|        | Receive data register                  | SCRDR1B      | 8              | H'FFFF8814 | 8              |
|        | Serial mode register                   | SCSMR1C      | 8              | H'FFFF9000 | 8              |
|        | Bit rate register                      | SCBRR1C      | 8              | H'FFFF9004 | 8              |
|        | Serial control register                | SCSCR1C      | 8              | H'FFFF9008 | 8              |
|        | Transmit data register                 | SCTDR1C      | 8              | H'FFFF900C | 8              |
|        | Serial status register                 | SCSSR1C      | 8              | H'FFFF9010 | 8              |
|        | Receive data register                  | SCRDR1C      | 8              | H'FFFF9014 | 8              |
|        | Serial mode register                   | SCSMR1D      | 8              | H'FFFF9800 | 8              |

| Module | Register Name                             | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|---|--------------|----------------|------------|----------------|
| SCI    | Bit rate register                         | SCBRR1D      | 8              | H'FFFF9804 | 8              |
|        | Serial control register                   | SCSCR1D      | 8              | H'FFFF9808 | 8              |
|        | Transmit data register                    | SCTDR1D      | 8              | H'FFFF980C | 8              |
|        | Serial status register                    | SCSSR1D      | 8              | H'FFFF9810 | 8              |
|        | Receive data register                     | SCRDR1D      | 8              | H'FFFF9814 | 8              |
|        | Serial mode register                      | SCSMR1E      | 8              | H'FFFFA000 | 8              |
|        | Bit rate register                         | SCBRR1E      | 8              | H'FFFFA004 | 8              |
|        | Serial control register                   | SCSCR1E      | 8              | H'FFFFA008 | 8              |
|        | Transmit data register                    | SCTDR1E      | 8              | H'FFFFA00C | 8              |
|        | Serial status register                    | SCSSR1E      | 8              | H'FFFFA010 | 8              |
|        | Receive data register                     | SCRDR1E      | 8              | H'FFFFA014 | 8              |
| RSPI   | RSPI control register A                   | SPCRA        | 8              | H'FFFFB000 | 8, 16          |
|        | RSPI slave select polarity register A     | SSLPA        | 8              | H'FFFFB001 | 8              |
|        | RSPI pin control register A               | SPPCRA       | 8              | H'FFFFB002 | 8, 16          |
|        | RSPI status register A                    | SPSRA        | 8              | H'FFFFB003 | 8              |
|        | RSPI data register A                      | SPDRA        | 16             | H'FFFFB004 | 16             |
|        | RSPI sequence control register A          | SPSCRA       | 8              | H'FFFFB008 | 8, 16          |
|        | RSPI sequence status register A           | SPSSRA       | 8              | H'FFFFB009 | 8              |
|        | RSPI bit rate register A                  | SPBRA        | 8              | H'FFFFB00A | 8              |
|        | RSPI clock delay register A               | SPCKDA       | 8              | H'FFFFB00C | 8, 16          |
|        | RSPI slave select negate delay register A | SSLNDA       | 8              | H'FFFFB00D | 8              |
|        | RSPI next-access delay register A         | SPNDA        | 8              | H'FFFFB00E | 8              |
|        | RSPI command register A0                  | SPCMDA0      | 16             | H'FFFFB010 | 16             |
|        | RSPI command register A1                  | SPCMDA1      | 16             | H'FFFFB012 | 16             |
|        | RSPI command register A2                  | SPCMDA2      | 16             | H'FFFFB014 | 16             |
|        | RSPI command register A3                  | SPCMDA3      | 16             | H'FFFFB016 | 16             |
|        | RSPI command register A4                  | SPCMDA4      | 16             | H'FFFFB018 | 16             |
|        | RSPI command register A5                  | SPCMDA5      | 16             | H'FFFFB01A | 16             |

| Module | Register Name                             | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------|---|--------------|----------------|------------|----------------|
| RSPI   | RSPI command register A6                  | SPCMDA6      | 16             | H'FFFFB01C | 16             |
|        | RSPI command register A7                  | SPCMDA7      | 16             | H'FFFFB01E | 16             |
|        | RSPI control register B                   | SPCRB        | 8              | H'FFFFB800 | 8, 16          |
|        | RSPI slave select polarity register B     | SSLPB        | 8              | H'FFFFB801 | 8              |
|        | RSPI pin control register B               | SPPCRB       | 8              | H'FFFFB802 | 8, 16          |
|        | RSPI status register B                    | SPSRB        | 8              | H'FFFFB803 | 8              |
|        | RSPI data register B                      | SPDRB        | 16             | H'FFFFB804 | 16             |
|        | RSPI sequence control register B          | SPSCRB       | 8              | H'FFFFB808 | 8, 16          |
|        | RSPI sequence status register B           | SPSSRB       | 8              | H'FFFFB809 | 8              |
|        | RSPI bit rate register B                  | SPBRB        | 8              | H'FFFFB80A | 8              |
|        | RSPI clock delay register B               | SPCKDB       | 8              | H'FFFFB80C | 8, 16          |
|        | RSPI slave select negate delay register B | SSLNDB       | 8              | H'FFFFB80D | 8              |
|        | RSPI next-access delay register B         | SPNDB        | 8              | H'FFFFB80E | 8              |
|        | RSPI command register B0                  | SPCMDB0      | 16             | H'FFFFB810 | 16             |
|        | RSPI command register B1                  | SPCMDB1      | 16             | H'FFFFB812 | 16             |
|        | RSPI command register B2                  | SPCMDB2      | 16             | H'FFFFB814 | 16             |
|        | RSPI command register B3                  | SPCMDB3      | 16             | H'FFFFB816 | 16             |
|        | RSPI command register B4                  | SPCMDB4      | 16             | H'FFFFB818 | 16             |
|        | RSPI command register B5                  | SPCMDB5      | 16             | H'FFFFB81A | 16             |
|        | RSPI command register B6                  | SPCMDB6      | 16             | H'FFFFB81C | 16             |
|        | RSPI command register B7                  | SPCMDB7      | 16             | H'FFFFB81E | 16             |
|        | RSPI control register C                   | SPCRC        | 8              | H'FFFFC000 | 8, 16          |
|        | RSPI slave select polarity register C     | SSLPC        | 8              | H'FFFFC001 | 8              |
|        | RSPI pin control register C               | SPPCRC       | 8              | H'FFFFC002 | 8, 16          |
|        | RSPI status register C                    | SPSRC        | 8              | H'FFFFC003 | 8              |
|        | RSPI data register C                      | SPDRC        | 16             | H'FFFFC004 | 16             |
|        | RSPI sequence control register C          | SPSCRC       | 8              | H'FFFFC008 | 8, 16          |

| Module        | Register Name                             | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|---|--------------|----------------|------------|----------------|
| RSPI          | RSPI sequence status register C           | SPSSRC       | 8              | H'FFFFC009 | 8              |
|               | RSPI bit rate register C                  | SPBRC        | 8              | H'FFFFC00A | 8              |
|               | RSPI clock delay register C               | SPCKDC       | 8              | H'FFFFC00C | 8, 16          |
|               | RSPI slave select negate delay register C | SSLNDC       | 8              | H'FFFFC00D | 8              |
|               | RSPI next-access delay register C         | SPNDC        | 8              | H'FFFFC00E | 8              |
|               | RSPI command register C0                  | SPCMDC0      | 16             | H'FFFFC010 | 16             |
|               | RSPI command register C1                  | SPCMDC1      | 16             | H'FFFFC012 | 16             |
|               | RSPI command register C2                  | SPCMDC2      | 16             | H'FFFFC014 | 16             |
|               | RSPI command register C3                  | SPCMDC3      | 16             | H'FFFFC016 | 16             |
|               | RSPI command register C4                  | SPCMDC4      | 16             | H'FFFFC018 | 16             |
|               | RSPI command register C5                  | SPCMDC5      | 16             | H'FFFFC01A | 16             |
|               | RSPI command register C6                  | SPCMDC6      | 16             | H'FFFFC01C | 16             |
|               | RSPI command register C7                  | SPCMDC7      | 16             | H'FFFFC01E | 16             |
| RCAN-         | Master control register                   | MCR          | 16             | H'FFFFD000 | 16             |
| TL1<br>(RCAN_ | General status register                   | GSR          | 16             | H'FFFFD002 | 16             |
| A)            | Bit configuration register 1              | BCR1         | 16             | H'FFFFD004 | 16             |
|               | Bit configuration register 0              | BCR0         | 16             | H'FFFFD006 | 16             |
|               | Interrupt register                        | IRR          | 16             | H'FFFFD008 | 16             |
|               | Interrupt mask register                   | IMR          | 16             | H'FFFFD00A | 16             |
|               | Error counter register                    | TEC/REC      | 16             | H'FFFFD00C | 16             |
|               | Transmit pending register 1               | TXPR1        | 16             | H'FFFFD020 | 32             |
|               | Transmit pending register 0               | TXPR0        | 16             | _          |                |
|               | Transmit cancel register 1                | TXCR1        | 16             | H'FFFFD028 | 16             |
|               | Transmit cancel register 0                | TXCR0        | 16             | H'FFFFD02A | 16             |
|               | Transmit acknowledge register 1           | TXACK1       | 16             | H'FFFFD030 | 16             |
|               | Transmit acknowledge register 0           | TXACK0       | 16             | H'FFFFD032 | 16             |
|               | Abort acknowledge register 1              | ABACK1       | 16             | H'FFFFD038 | 16             |
|               | Abort acknowledge register 0              | ABACK0       | 16             | H'FFFFD03A | 16             |

| Module       | Register                | Name                       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------------|-------------------------|----------------------------|--------------|----------------|------------|----------------|
| RCAN-<br>TL1 | Data fram<br>register 1 | ne receive pending         | RXPR1        | 16             | H'FFFFD040 | 16             |
| (RCAN_<br>A) | Data fram<br>register 0 | ne receive pending         | RXPR0        | 16             | H'FFFFD042 | 16             |
|              | Remote for register 1   | rame receive pending       | RFPR1        | 16             | H'FFFFD048 | 16             |
|              | Remote for register 0   | rame receive pending       | RFPR0        | 16             | H'FFFFD04A | 16             |
|              | Mailbox in              | nterrupt mask register 1   | MBIMR1       | 16             | H'FFFFD050 | 16             |
|              | Mailbox in              | nterrupt mask register 0   | MBIMR0       | 16             | H'FFFFD052 | 16             |
|              | Unread m                | nessage status             | UMSR1        | 16             | H'FFFFD058 | 16             |
|              | Unread m<br>register 0  | nessage status             | UMSR0        | 16             | H'FFFFD05A | 16             |
|              | Timer trig              | ger control register 0     | TTCR0        | 16             | H'FFFFD080 | 16             |
|              | Cycle ma<br>window re   | ximum/tx-enable<br>egister | CMAX_TEW     | 16             | H'FFFFD084 | 16             |
|              | Reference               | e trigger offset register  | RFTROFF      | 16             | H'FFFFD086 | 16             |
|              | Timer sta               | tus register               | TSR          | 16             | H'FFFFD088 | 16             |
|              | Cycle cou               | ınter register             | CCR          | 16             | H'FFFFD08A | 16             |
|              | Timer cou               | unter register             | TCNTR        | 16             | H'FFFFD08C | 16             |
|              | Cycle tim               | e register                 | CYCTR        | 16             | H'FFFFD090 | 16             |
|              | Reference               | e mark register            | RFMK         | 16             | H'FFFFD094 | 16             |
|              | Timer cor               | mpare match register 0     | TCMR0        | 16             | H'FFFFD098 | 16             |
|              | Timer cor               | mpare match register 1     | TCMR1        | 16             | H'FFFFD09C | 16             |
|              | Timer cor               | mpare match register 2     | TCMR2        | 16             | H'FFFFD0A0 | 16             |
|              | Tx-trigger              | time selection register    | TTTSEL       | 16             | H'FFFFD0A4 | 16             |
|              | MB[0].                  | CONTROL0_H                 | _            | 16             | H'FFFFD100 | 16, 32         |
|              |                         | CONTROL0_L                 | _            | 16             | H'FFFFD102 | 16             |
|              |                         | LAFM0                      | _            | 16             | H'FFFFD104 | 16, 32         |
|              |                         | LAFM1                      |              | 16             | H'FFFFD106 | 16             |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[0].   | DATA_01    | _            | 16             | H'FFFFD108 | 8, 16, 32      |
| TL1<br>(RCAN_ |          | DATA_23    | _            | 16             | H'FFFFD10A | 8, 16          |
| (110AN_       |          | DATA_45    | _            | 16             | H'FFFFD10C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD10E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD110 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFD112 | 16             |
|               | MB[1].   | CONTROL0_H | _            | 16             | H'FFFFD120 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD122 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD124 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD126 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFD128 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFD12A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFD12C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD12E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD130 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFD132 | 16             |
|               | MB[2].   | CONTROL0_H | _            | 16             | H'FFFFD140 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD142 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD144 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD146 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFD148 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFD14A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFD14C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD14E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD150 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFD152 | 16             |
|               | MB[3].   | CONTROL0_H | _            | 16             | H'FFFFD160 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD162 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD164 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD166 | 16             |
|               |          |            |              |                |            |                |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[3].   | DATA_01    | _            | 16             | H'FFFFD168 | 8, 16, 32      |
| TL1<br>(RCAN_ |          | DATA_23    | _            | 16             | H'FFFFD16A | 8, 16          |
| A)            |          | DATA_45    | _            | 16             | H'FFFFD16C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD16E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD170 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFD172 | 16             |
|               | MB[4].   | CONTROL0_H | _            | 16             | H'FFFFD180 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD182 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD184 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD186 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFD188 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFD18A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFD18C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD18E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD190 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFD192 | 16             |
|               | MB[5].   | CONTROL0_H | _            | 16             | H'FFFFD1A0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD1A2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD1A4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD1A6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFD1A8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFD1AA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFD1AC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD1AE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD1B0 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFD1B2 | 16             |
|               | MB[6].   | CONTROL0_H | _            | 16             | H'FFFFD1C0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD1C2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD1C4 | 16, 32         |
| ·             |          | LAFM1      | _            | 16             | H'FFFFD1C6 | 16             |

| RCAN- TL1 (RCAN_ A)  MB[6]. DATA_01 — 16 H'FFFD1C8 8, 16, 3  DATA_23 — 16 H'FFFD1CA 8, 16  DATA_45 — 16 H'FFFFD1CE 8, 16, 3  DATA_67 — 16 H'FFFFD1CE 8, 16  CONTROL1 — 16 H'FFFFD1D0 8, 16  TIMESTAMP — 16 H'FFFFD1D2 16  MB[7]. CONTROL0_H — 16 H'FFFFD1E0 16, 32  CONTROL0_L — 16 H'FFFFD1E2 16  LAFM0 — 16 H'FFFFD1E4 16, 32  LAFM1 — 16 H'FFFFD1E6 16  DATA_01 — 16 H'FFFFD1E8 8, 16, 3  DATA_23 — 16 H'FFFFD1EA 8, 16  DATA_45 — 16 H'FFFFD1EC 8, 16, 3  DATA_45 — 16 H'FFFFD1EC 8, 16, 3                          | ; |
|---|---|
| CONTROLO_H  | 2 |
| A) DATA_45 — 16 H'FFFD1CC 8, 16, 3  DATA_67 — 16 H'FFFD1CE 8, 16  CONTROL1 — 16 H'FFFD1D0 8, 16  TIMESTAMP — 16 H'FFFD1D2 16  MB[7]. CONTROL0_H — 16 H'FFFD1E0 16, 32  CONTROL0_L — 16 H'FFFD1E2 16  LAFM0 — 16 H'FFFD1E4 16, 32  LAFM1 — 16 H'FFFD1E6 16  DATA_01 — 16 H'FFFD1E8 8, 16, 3  DATA_23 — 16 H'FFFD1EA 8, 16  DATA_45 — 16 H'FFFD1EC 8, 16, 3   |   |
| CONTROL1 — 16 H'FFFD1D0 8, 16  TIMESTAMP — 16 H'FFFD1D2 16  MB[7]. CONTROL0_H — 16 H'FFFD1E0 16, 32  CONTROL0_L — 16 H'FFFD1E2 16  LAFM0 — 16 H'FFFD1E4 16, 32  LAFM1 — 16 H'FFFD1E6 16  DATA_01 — 16 H'FFFD1E8 8, 16, 3  DATA_23 — 16 H'FFFD1EA 8, 16  DATA_45 — 16 H'FFFD1EC 8, 16, 3   | 2 |
| TIMESTAMP       —       16       H'FFFFD1D2       16         MB[7].       CONTROLO_H       —       16       H'FFFFD1E0       16, 32         CONTROLO_L       —       16       H'FFFFD1E2       16         LAFM0       —       16       H'FFFFD1E4       16, 32         LAFM1       —       16       H'FFFFD1E6       16         DATA_01       —       16       H'FFFFD1E8       8, 16, 3         DATA_23       —       16       H'FFFFD1EA       8, 16         DATA_45       —       16       H'FFFFD1EC       8, 16, 3 |   |
| MB[7]. CONTROLO_H — 16 H'FFFD1E0 16, 32  CONTROLO_L — 16 H'FFFD1E2 16  LAFM0 — 16 H'FFFD1E4 16, 32  LAFM1 — 16 H'FFFD1E6 16  DATA_01 — 16 H'FFFD1E8 8, 16, 3  DATA_23 — 16 H'FFFD1EA 8, 16  DATA_45 — 16 H'FFFD1EC 8, 16, 3   |   |
| CONTROLO_L — 16 H'FFFD1E2 16  LAFM0 — 16 H'FFFFD1E4 16, 32  LAFM1 — 16 H'FFFFD1E6 16  DATA_01 — 16 H'FFFFD1E8 8, 16, 3  DATA_23 — 16 H'FFFFD1EA 8, 16  DATA_45 — 16 H'FFFFD1EC 8, 16, 3   |   |
| LAFM0       —       16       H'FFFFD1E4       16, 32         LAFM1       —       16       H'FFFFD1E6       16         DATA_01       —       16       H'FFFFD1E8       8, 16, 3         DATA_23       —       16       H'FFFFD1EA       8, 16         DATA_45       —       16       H'FFFFD1EC       8, 16, 3   |   |
| LAFM1       —       16       H'FFFFD1E6       16         DATA_01       —       16       H'FFFFD1E8       8, 16, 3         DATA_23       —       16       H'FFFFD1EA       8, 16         DATA_45       —       16       H'FFFFD1EC       8, 16, 3  |   |
| DATA_01       —       16       H'FFFFD1E8       8, 16, 3         DATA_23       —       16       H'FFFFD1EA       8, 16         DATA_45       —       16       H'FFFFD1EC       8, 16, 3   |   |
| DATA_23         —         16         H'FFFFD1EA         8, 16           DATA_45         —         16         H'FFFFD1EC         8, 16, 3  |   |
| DATA_45 — 16 H'FFFFD1EC 8, 16, 3  | 2 |
|   |   |
| DATA_67 — 16 H'FFFFD1EE 8, 16   | 2 |
|   |   |
| CONTROL1 — 16 H'FFFFD1F0 8, 16  |   |
| TIMESTAMP — 16 H'FFFFD1F2 16  |   |
| MB[8]. CONTROL0_H 16 H'FFFFD200 16, 32  |   |
| CONTROLO_L — 16 H'FFFFD202 16   |   |
| LAFM0 — 16 H'FFFFD204 16, 32  |   |
| LAFM1 — 16 H'FFFFD206 16  |   |
| DATA_01 — 16 H'FFFFD208 8, 16, 3  | 2 |
| DATA_23 — 16 H'FFFFD20A 8, 16   |   |
| DATA_45 — 16 H'FFFFD20C 8, 16, 3  | 2 |
| DATA_67 — 16 H'FFFFD20E 8, 16   |   |
| CONTROL1 — 16 H'FFFFD210 8, 16  |   |
| TIMESTAMP — 16 H'FFFFD212 16  |   |
| MB[9]. CONTROL0_H — 16 H'FFFFD220 16, 32  |   |
| CONTROLO_L — 16 H'FFFFD222 16   | _ |
| LAFM0 — 16 H'FFFFD224 16, 32  |   |
| LAFM1 — 16 H'FFFFD226 16  |   |

| Module         | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|----------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-          | MB[9].   | DATA_01    | _            | 16             | H'FFFFD228 | 8, 16, 32      |
| TL1<br>(RCAN_  |          | DATA_23    | _            | 16             | H'FFFFD22A | 8, 16          |
| (110AIN_<br>A) |          | DATA_45    | _            | 16             | H'FFFFD22C | 8, 16, 32      |
|                |          | DATA_67    | _            | 16             | H'FFFFD22E | 8, 16          |
|                |          | CONTROL1   | _            | 16             | H'FFFFD230 | 8, 16          |
|                |          | TIMESTAMP  | _            | 16             | H'FFFFD232 | 16             |
|                | MB[10].  | CONTROL0_H | _            | 16             | H'FFFFD240 | 16, 32         |
|                |          | CONTROL0_L | _            | 16             | H'FFFFD242 | 16             |
|                |          | LAFM0      | _            | 16             | H'FFFFD244 | 16, 32         |
|                |          | LAFM1      | _            | 16             | H'FFFFD246 | 16             |
|                |          | DATA_01    | _            | 16             | H'FFFFD248 | 8, 16, 32      |
|                |          | DATA_23    | _            | 16             | H'FFFFD24A | 8, 16          |
|                |          | DATA_45    | _            | 16             | H'FFFFD24C | 8, 16, 32      |
|                |          | DATA_67    | _            | 16             | H'FFFFD24E | 8, 16          |
|                |          | CONTROL1   | _            | 16             | H'FFFFD250 | 8, 16          |
|                |          | TIMESTAMP  | _            | 16             | H'FFFFD252 | 16             |
|                | MB[11].  | CONTROL0_H | _            | 16             | H'FFFFD260 | 16, 32         |
|                |          | CONTROL0_L | _            | 16             | H'FFFFD262 | 16             |
|                |          | LAFM0      | _            | 16             | H'FFFFD264 | 16, 32         |
|                |          | LAFM1      | _            | 16             | H'FFFFD266 | 16             |
|                |          | DATA_01    | _            | 16             | H'FFFFD268 | 8, 16, 32      |
|                |          | DATA_23    | _            | 16             | H'FFFFD26A | 8, 16          |
|                |          | DATA_45    | _            | 16             | H'FFFFD26C | 8, 16, 32      |
|                |          | DATA_67    | _            | 16             | H'FFFFD26E | 8, 16          |
|                |          | CONTROL1   | _            | 16             | H'FFFFD270 | 8, 16          |
|                |          | TIMESTAMP  | _            | 16             | H'FFFFD272 | 16             |
|                | MB[12].  | CONTROL0_H | _            | 16             | H'FFFFD280 | 16, 32         |
|                |          | CONTROL0_L | _            | 16             | H'FFFFD282 | 16             |
|                |          | LAFM0      | _            | 16             | H'FFFFD284 | 16, 32         |
|                |          | LAFM1      |              | 16             | H'FFFFD286 | 16             |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[12].  | DATA_01    | _            | 16             | H'FFFFD288 | 8, 16, 32      |
| TL1<br>(RCAN_ |          | DATA_23    | <del>_</del> | 16             | H'FFFFD28A | 8, 16          |
| A)            |          | DATA_45    | _            | 16             | H'FFFFD28C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD28E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD290 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFD292 | 16             |
|               | MB[13].  | CONTROL0_H | _            | 16             | H'FFFFD2A0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD2A2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD2A4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD2A6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFD2A8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFD2AA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFD2AC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD2AE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD2B0 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFD2B2 | 16             |
|               | MB[14].  | CONTROL0_H | _            | 16             | H'FFFFD2C0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD2C2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD2C4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD2C6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFD2C8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFD2CA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFD2CC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD2CE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD2D0 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFD2D2 | 16             |
|               | MB[15].  | CONTROL0_H | _            | 16             | H'FFFFD2E0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD2E2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD2E4 | 16, 32         |
|               |          | LAFM1      |              | 16             | H'FFFFD2E6 | 16             |

| RCAN-         MB[15].         DATA_01         —         16         H'FFFFD2E8           TL1         DATA_23         —         16         H'FFFFD2EA | 8, 16     |
|---|-----------|
|   |           |
| (RCAN_  | 0 16 00   |
| A) DATA_45 — 16 H'FFFFD2EC  | 0, 10, 32 |
| DATA_67 — 16 H'FFFFD2EE   | 8, 16     |
| CONTROL1 — 16 H'FFFFD2F0  | 8, 16     |
| TIMESTAMP — 16 H'FFFFD2F2   | 16        |
| MB[16]. CONTROL0_H — 16 H'FFFFD300  | 16, 32    |
| CONTROLO_L — 16 H'FFFFD302  | 16        |
| LAFM0 — 16 H'FFFFD304   | 16, 32    |
| LAFM1 — 16 H'FFFFD306   | 16        |
| DATA_01 — 16 H'FFFFD308   | 8, 16, 32 |
| DATA_23 — 16 H'FFFFD30A   | 8, 16     |
| DATA_45 — 16 H'FFFFD30C   | 8, 16, 32 |
| DATA_67 — 16 H'FFFFD30E   | 8, 16     |
| CONTROL1 — 16 H'FFFFD310  | 8, 16     |
| MB[17]. CONTROL0_H — 16 H'FFFFD320  | 16, 32    |
| CONTROLO_L — 16 H'FFFFD322  | 16        |
| LAFM0 — 16 H'FFFFD324   | 16, 32    |
| LAFM1 — 16 H'FFFFD326   | 16        |
| DATA_01 — 16 H'FFFFD328   | 8, 16, 32 |
| DATA_23 — 16 H'FFFFD32A   | 8, 16     |
| DATA_45 — 16 H'FFFFD32C   | 8, 16, 32 |
| DATA_67 — 16 H'FFFFD32E   | 8, 16     |
| CONTROL1 — 16 H'FFFFD330  | 8, 16     |
| MB[18]. CONTROL0_H — 16 H'FFFFD340  | 16, 32    |
| CONTROLO_L — 16 H'FFFFD342  | 16        |
| LAFM0 — 16 H'FFFFD344   | 16, 32    |
| LAFM1 — 16 H'FFFFD346   | 16        |
| DATA_01 — 16 H'FFFFD348   | 8, 16, 32 |
| DATA_23 — 16 H'FFFFD34A   | 8, 16     |

| RCAN-TL1 (RCAN_A)   | Module | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---|--------|----------|------------|--------------|----------------|------------|----------------|
| RCAN_A    A    B    A    A    A    A    A   | _      | MB[18].  | DATA_45    | _            | 16             | H'FFFFD34C | 8, 16, 32      |
| A) CONTROL1 — 16 HFFFFD350 8, 16  MB[19]. CONTROLO_H — 16 HFFFFD360 16, 32  CONTROLO_L — 16 HFFFFD362 16  LAFM0 — 16 HFFFFD364 16, 32  LAFM1 — 16 HFFFFD366 16  DATA_01 — 16 HFFFFD368 8, 16, 32  DATA_23 — 16 HFFFFD368 8, 16  DATA_67 — 16 HFFFFD36E 8, 16  CONTROL1 — 16 HFFFFD36 8, 16  MB[20]. CONTROLO_H — 16 HFFFFD380 16, 32  CONTROLO_L — 16 HFFFFD380 16, 32  LAFM1 — 16 HFFFFD380 16, 32  LAFM1 — 16 HFFFFD380 16, 32  LAFM1 — 16 HFFFFD380 16, 32  DATA_23 — 16 HFFFFD380 16, 32  DATA_23 — 16 HFFFFD380 16  DATA_01 — 16 HFFFFD380 8, 16  DATA_23 — 16 HFFFFD380 8, 16  DATA_45 — 16 HFFFFD380 8, 16  DATA_67 — 16 HFFFFD380 8, 16  MB[21]. CONTROLO_H — 16 HFFFFD380 8, 16  CONTROLO_L — 16 HFFFFD380 16, 32  DATA_67 — 16 HFFFFD380 16, 32  LAFM1 — 16 HFFFFD380 8, 16  DATA_01 — 16 HFFFFD380 16, 32  DATA_67 — 16 HFFFFD380 16, 32  LAFM1 — 16 HFFFFD380 16, 32  LAFM1 — 16 HFFFFD380 16, 32  DATA_01 — 16 HFFFFD380 16, 32  LAFM1 — 16 HFFFFD380 16, 32  DATA_01 — 16 HFFFFD380 8, 16  DATA_01 — 16 HFFFFD380 8, 16 |        |          | DATA_67    | _            | 16             | H'FFFFD34E | 8, 16          |
| CONTROLO_L — 16 H'FFFD362 16  LAFM0 — 16 H'FFFD364 16, 32  LAFM1 — 16 H'FFFD366 16  DATA_01 — 16 H'FFFD368 8, 16, 32  DATA_23 — 16 H'FFFD368 8, 16  DATA_45 — 16 H'FFFD36C 8, 16, 32  DATA_67 — 16 H'FFFD36C 8, 16  CONTROL1 — 16 H'FFFD36C 8, 16  CONTROL0_H — 16 H'FFFD380 16, 32  CONTROL0_L — 16 H'FFFD382 16  LAFM0 — 16 H'FFFD386 16  DATA_23 — 16 H'FFFD386 8, 16, 32  LAFM1 — 16 H'FFFD386 8, 16, 32  DATA_23 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD38C 8, 16, 32  MB[21]. CONTROLO_H — 16 H'FFFD38C 8, 16, 32  DATA_01 — 16 H'FFFD3AC 16, 32  CONTROL0_H — 16 H'FFFD3AC 16, 32  CONTROLO_L — 16 H'FFFD3AC 16, 32  LAFM1 — 16 H'FFFD3AC 16, 32  LAFM1 — 16 H'FFFD3AC 16, 32  DATA_01 — 16 H'FFFD3AC 8, 16, 32  |        |          | CONTROL1   | _            | 16             | H'FFFFD350 | 8, 16          |
| LAFM0 — 16 H'FFFD364 16, 32  LAFM1 — 16 H'FFFD366 16  DATA_01 — 16 H'FFFD368 8, 16, 32  DATA_23 — 16 H'FFFD36A 8, 16  DATA_45 — 16 H'FFFD36C 8, 16, 32  DATA_67 — 16 H'FFFD36C 8, 16  CONTROL1 — 16 H'FFFD36 16, 32  CONTROL0_H — 16 H'FFFD382 16  LAFM0 — 16 H'FFFD386 16, 32  LAFM1 — 16 H'FFFD38A 8, 16  DATA_23 — 16 H'FFFD38B 8, 16, 32  DATA_67 — 16 H'FFFD38C 8, 16  DATA_23 — 16 H'FFFD38C 8, 16  DATA_67 — 16 H'FFFD38C 8, 16  CONTROL1 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD38C 8, 16  CONTROL1 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD3AC 16, 32  CONTROL0_L — 16 H'FFFD3AC 16, 32  CONTROL0_L — 16 H'FFFD3AC 16, 32  LAFM1 — 16 H'FFFD3AC 16, 32  DATA_23 — 16 H'FFFFD3AC 16, 32  DATA_01 — 16 H'FFFD3AC 16, 32  DATA_01 — 16 H'FFFFD3AC 16, 32  DATA_01 — 16 H'FFFD3AC 16, 32  DATA_01 — 16 H'FFFD3AC 16, 32  DATA_23 — 16 H'FFFFD3AC 16, 32  DATA_45 — 16 H'FFFFD3AC 16, 32   |        | MB[19].  | CONTROL0_H | _            | 16             | H'FFFFD360 | 16, 32         |
| LAFM1   |        |          | CONTROL0_L | _            | 16             | H'FFFFD362 | 16             |
| DATA_01         —         16         H'FFFFD368         8, 16, 32           DATA_23         —         16         H'FFFFD36A         8, 16           DATA_45         —         16         H'FFFFD36C         8, 16, 32           DATA_67         —         16         H'FFFD36E         8, 16           CONTROL1         —         16         H'FFFD370         8, 16           CONTROL0_H         —         16         H'FFFD380         16, 32           CONTROL0_L         —         16         H'FFFFD382         16           LAFM0         —         16         H'FFFFD384         16, 32           LAFM1         —         16         H'FFFFD386         16           DATA_01         —         16         H'FFFFD388         8, 16, 32           DATA_23         —         16         H'FFFFD388         8, 16, 32           DATA_45         —         16         H'FFFFD38C         8, 16, 32           DATA_67         —         16         H'FFFFD3A0         16, 32           CONTROL0_H         —         16         H'FFFFD3A2         16           LAFM0         —         16         H'FFFFD3A2         16   |        |          | LAFM0      | _            | 16             | H'FFFFD364 | 16, 32         |
| DATA_23 — 16 H'FFFD36A 8, 16  DATA_45 — 16 H'FFFD36C 8, 16, 32  DATA_67 — 16 H'FFFD36E 8, 16  CONTROL1 — 16 H'FFFD370 8, 16  MB[20]. CONTROL0_H — 16 H'FFFD382 16  LAFM0 — 16 H'FFFD386 16  DATA_01 — 16 H'FFFD38A 8, 16  DATA_23 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD38E 8, 16  MB[21]. CONTROL0_H — 16 H'FFFD38A 8, 16  DATA_01 — 16 H'FFFD38A 8, 16  DATA_01 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD38C 8, 16  CONTROL1 — 16 H'FFFD38C 8, 16  AMB[21]. CONTROL0_H — 16 H'FFFD3AC 16, 32  CONTROL0_L — 16 H'FFFD3AC 16, 32  LAFM1 — 16 H'FFFD3AC 16, 32  LAFM1 — 16 H'FFFD3AC 16  LAFM0 — 16 H'FFFD3AC 16  LAFM0 — 16 H'FFFD3AC 16  DATA_01 — 16 H'FFFD3AC 8, 16, 32  DATA_23 — 16 H'FFFD3AC 8, 16, 32  DATA_23 — 16 H'FFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFD3AC 8, 16, 32   |        |          | LAFM1      | _            | 16             | H'FFFFD366 | 16             |
| DATA_45         —         16         H'FFFFD36C         8, 16, 32           DATA_67         —         16         H'FFFFD36E         8, 16           CONTROL1         —         16         H'FFFFD370         8, 16           MB[20].         CONTROL0_H         —         16         H'FFFFD380         16, 32           CONTROL0_L         —         16         H'FFFFD382         16           LAFM0         —         16         H'FFFFD384         16, 32           LAFM1         —         16         H'FFFFD386         16           DATA_01         —         16         H'FFFFD388         8, 16, 32           DATA_23         —         16         H'FFFFD38C         8, 16, 32           DATA_45         —         16         H'FFFFD38E         8, 16           CONTROL1         —         16         H'FFFFD30         8, 16           MB[21].         CONTROL0_H         —         16         H'FFFFD3A2         16           LAFM0         —         16         H'FFFFD3A2         16           LAFM1         —         16         H'FFFFD3A6         16           DATA_23         —         16         H'FFFFD3AA         8,   |        |          | DATA_01    | _            | 16             | H'FFFFD368 | 8, 16, 32      |
| DATA_67         —         16         H'FFFD36E         8, 16           CONTROL1         —         16         H'FFFD370         8, 16           MB[20].         CONTROL0_H         —         16         H'FFFFD380         16, 32           CONTROL0_L         —         16         H'FFFFD382         16           LAFM0         —         16         H'FFFFD384         16, 32           LAFM1         —         16         H'FFFFD386         16           DATA_01         —         16         H'FFFFD388         8, 16, 32           DATA_23         —         16         H'FFFFD38C         8, 16, 32           DATA_45         —         16         H'FFFFD38C         8, 16, 32           DATA_67         —         16         H'FFFFD38C         8, 16           CONTROL1         —         16         H'FFFFD3AC         16, 32           CONTROL0_H         —         16         H'FFFFD3AC         16, 32           LAFM0         —         16         H'FFFFD3AC         16, 32           LAFM1         —         16         H'FFFFD3AC         8, 16, 32           DATA_23         —         16         H'FFFFD3AC         8, 16, 32 <td></td> <td></td> <td>DATA_23</td> <td>_</td> <td>16</td> <td>H'FFFFD36A</td> <td>8, 16</td>  |        |          | DATA_23    | _            | 16             | H'FFFFD36A | 8, 16          |
| CONTROL1         —         16         H'FFFFD370         8, 16           MB[20].         CONTROL0_H         —         16         H'FFFFD380         16, 32           CONTROL0_L         —         16         H'FFFFD382         16           LAFM0         —         16         H'FFFFD384         16, 32           LAFM1         —         16         H'FFFFD386         16           DATA_01         —         16         H'FFFFD388         8, 16, 32           DATA_23         —         16         H'FFFFD38C         8, 16, 32           DATA_45         —         16         H'FFFFD38C         8, 16, 32           DATA_67         —         16         H'FFFFD38E         8, 16           CONTROL1         —         16         H'FFFFD3A0         16, 32           CONTROL0_H         —         16         H'FFFFD3A2         16           LAFM0         —         16         H'FFFFD3A6         16           LAFM1         —         16         H'FFFFD3A8         8, 16, 32           DATA_01         —         16         H'FFFFD3AC         8, 16, 32           DATA_45         —         16         H'FFFFD3AC         8, 16, 32   |        |          | DATA_45    | _            | 16             | H'FFFFD36C | 8, 16, 32      |
| MB[20]. CONTROLO_H — 16 H'FFFD380 16, 32  CONTROLO_L — 16 H'FFFD382 16  LAFM0 — 16 H'FFFD384 16, 32  LAFM1 — 16 H'FFFD386 16  DATA_01 — 16 H'FFFD388 8, 16, 32  DATA_23 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD38E 8, 16  CONTROL1 — 16 H'FFFD380 8, 16  MB[21]. CONTROLO_H — 16 H'FFFD3A0 16, 32  CONTROLO_L — 16 H'FFFD3A2 16  LAFM0 — 16 H'FFFD3A2 16  LAFM1 — 16 H'FFFD3A6 16  DATA_01 — 16 H'FFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFD3AA 8, 16  DATA_23 — 16 H'FFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFFD3AC 8, 16, 32  DATA_67 — 16 H'FFFFD3AC 8, 16, 32   |        |          | DATA_67    | _            | 16             | H'FFFFD36E | 8, 16          |
| CONTROLO_L — 16 H'FFFD382 16  LAFM0 — 16 H'FFFFD384 16, 32  LAFM1 — 16 H'FFFFD386 16  DATA_01 — 16 H'FFFFD388 8, 16, 32  DATA_23 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD38E 8, 16  CONTROL1 — 16 H'FFFD38C 8, 16, 32  MB[21]. CONTROLO_H — 16 H'FFFD3A0 16, 32  CONTROLO_L — 16 H'FFFD3A2 16  LAFM0 — 16 H'FFFD3A4 16, 32  LAFM1 — 16 H'FFFD3A6 16  DATA_01 — 16 H'FFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFD3AA 8, 16  DATA_23 — 16 H'FFFD3AA 8, 16  DATA_23 — 16 H'FFFD3AC 8, 16, 32  DATA_67 — 16 H'FFFD3AC 8, 16, 32  |        |          | CONTROL1   | _            | 16             | H'FFFFD370 | 8, 16          |
| LAFM0 — 16 H'FFFD384 16, 32  LAFM1 — 16 H'FFFD386 16  DATA_01 — 16 H'FFFD388 8, 16, 32  DATA_23 — 16 H'FFFD38C 8, 16, 32  DATA_45 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD38E 8, 16  CONTROL1 — 16 H'FFFD390 8, 16  MB[21]. CONTROL0_H — 16 H'FFFD3A0 16, 32  CONTROL0_L — 16 H'FFFD3A2 16  LAFM0 — 16 H'FFFD3A4 16, 32  LAFM1 — 16 H'FFFD3A6 16  DATA_01 — 16 H'FFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFD3AA 8, 16  DATA_45 — 16 H'FFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFD3AC 8, 16, 32   |        | MB[20].  | CONTROL0_H | _            | 16             | H'FFFFD380 | 16, 32         |
| LAFM1 — 16 H'FFFD386 16  DATA_01 — 16 H'FFFFD388 8, 16, 32  DATA_23 — 16 H'FFFFD38A 8, 16  DATA_45 — 16 H'FFFFD38C 8, 16, 32  DATA_67 — 16 H'FFFFD38E 8, 16  CONTROL1 — 16 H'FFFFD39O 8, 16  MB[21]. CONTROL0_H — 16 H'FFFFD3A2 16  LAFM0 — 16 H'FFFFD3A2 16  DATA_01 — 16 H'FFFFD3A6 16  DATA_01 — 16 H'FFFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFFD3AC 8, 16, 32  DATA_67 — 16 H'FFFFD3AC 8, 16, 32  |        |          | CONTROL0_L | _            | 16             | H'FFFFD382 | 16             |
| DATA_01 — 16 H'FFFD388 8, 16, 32  DATA_23 — 16 H'FFFD38A 8, 16  DATA_45 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD38E 8, 16  CONTROL1 — 16 H'FFFD390 8, 16  MB[21]. CONTROL0_H — 16 H'FFFD3A2 16, 32  CONTROL0_L — 16 H'FFFD3A2 16  LAFM0 — 16 H'FFFD3A4 16, 32  LAFM1 — 16 H'FFFD3A6 16  DATA_01 — 16 H'FFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFD3AC 8, 16, 32  DATA_67 — 16 H'FFFD3AE 8, 16   |        |          | LAFM0      | _            | 16             | H'FFFFD384 | 16, 32         |
| DATA_23 — 16 H'FFFD38A 8, 16  DATA_45 — 16 H'FFFD38C 8, 16, 32  DATA_67 — 16 H'FFFD38E 8, 16  CONTROL1 — 16 H'FFFD39O 8, 16  MB[21]. CONTROL0_H — 16 H'FFFD3AO 16, 32  CONTROL0_L — 16 H'FFFD3A2 16  LAFMO — 16 H'FFFD3A4 16, 32  LAFM1 — 16 H'FFFD3A6 16  DATA_01 — 16 H'FFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFD3AC 8, 16, 32  DATA_67 — 16 H'FFFD3AE 8, 16   |        |          | LAFM1      | _            | 16             | H'FFFFD386 | 16             |
| DATA_45 — 16 H'FFFFD38C 8, 16, 32  DATA_67 — 16 H'FFFFD38E 8, 16  CONTROL1 — 16 H'FFFFD390 8, 16  MB[21]. CONTROL0_H — 16 H'FFFFD3A0 16, 32  CONTROL0_L — 16 H'FFFFD3A2 16  LAFM0 — 16 H'FFFFD3A4 16, 32  LAFM1 — 16 H'FFFFD3A6 16  DATA_01 — 16 H'FFFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFFD3AC 8, 16, 32  DATA_45 — 16 H'FFFFD3AE 8, 16   |        |          | DATA_01    | _            | 16             | H'FFFFD388 | 8, 16, 32      |
| DATA_67 — 16 H'FFFD38E 8, 16  CONTROL1 — 16 H'FFFFD390 8, 16  MB[21]. CONTROL0_H — 16 H'FFFFD3A0 16, 32  CONTROL0_L — 16 H'FFFFD3A2 16  LAFM0 — 16 H'FFFFD3A4 16, 32  LAFM1 — 16 H'FFFFD3A6 16  DATA_01 — 16 H'FFFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFFD3AA 8, 16  DATA_45 — 16 H'FFFFD3AE 8, 16   |        |          | DATA_23    | _            | 16             | H'FFFFD38A | 8, 16          |
| CONTROL1 — 16 H'FFFD390 8, 16  MB[21]. CONTROL0_H — 16 H'FFFFD3A0 16, 32  CONTROL0_L — 16 H'FFFFD3A2 16  LAFM0 — 16 H'FFFFD3A4 16, 32  LAFM1 — 16 H'FFFFD3A6 16  DATA_01 — 16 H'FFFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFFD3AA 8, 16  DATA_45 — 16 H'FFFFD3AC 8, 16, 32  DATA_67 — 16 H'FFFFD3AE 8, 16   |        |          | DATA_45    | _            | 16             | H'FFFFD38C | 8, 16, 32      |
| MB[21]. CONTROLO_H — 16 H'FFFFD3A0 16, 32  CONTROLO_L — 16 H'FFFFD3A2 16  LAFM0 — 16 H'FFFFD3A4 16, 32  LAFM1 — 16 H'FFFFD3A6 16  DATA_01 — 16 H'FFFFD3A8 8, 16, 32  DATA_23 — 16 H'FFFFD3AA 8, 16  DATA_45 — 16 H'FFFFD3AC 8, 16, 32  DATA_67 — 16 H'FFFFD3AE 8, 16  |        |          | DATA_67    | _            | 16             | H'FFFFD38E | 8, 16          |
| CONTROLO_L       —       16       H'FFFFD3A2       16         LAFM0       —       16       H'FFFFD3A4       16, 32         LAFM1       —       16       H'FFFFD3A6       16         DATA_01       —       16       H'FFFFD3A8       8, 16, 32         DATA_23       —       16       H'FFFFD3AC       8, 16, 32         DATA_45       —       16       H'FFFFD3AC       8, 16, 32         DATA_67       —       16       H'FFFFD3AE       8, 16   |        |          | CONTROL1   | _            | 16             | H'FFFFD390 | 8, 16          |
| LAFM0       —       16       H'FFFFD3A4       16, 32         LAFM1       —       16       H'FFFFD3A6       16         DATA_01       —       16       H'FFFFD3A8       8, 16, 32         DATA_23       —       16       H'FFFFD3AA       8, 16         DATA_45       —       16       H'FFFFD3AC       8, 16, 32         DATA_67       —       16       H'FFFFD3AE       8, 16   |        | MB[21].  | CONTROL0_H | _            | 16             | H'FFFFD3A0 | 16, 32         |
| LAFM1       —       16       H'FFFFD3A6       16         DATA_01       —       16       H'FFFFD3A8       8, 16, 32         DATA_23       —       16       H'FFFFD3AA       8, 16         DATA_45       —       16       H'FFFFD3AC       8, 16, 32         DATA_67       —       16       H'FFFFD3AE       8, 16  |        |          | CONTROL0_L | _            | 16             | H'FFFFD3A2 | 16             |
| DATA_01       —       16       H'FFFFD3A8 8, 16, 32         DATA_23       —       16       H'FFFFD3AA 8, 16         DATA_45       —       16       H'FFFFD3AC 8, 16, 32         DATA_67       —       16       H'FFFFD3AE 8, 16   |        |          | LAFM0      | _            | 16             | H'FFFFD3A4 | 16, 32         |
| DATA_23       —       16       H'FFFFD3AA 8, 16         DATA_45       —       16       H'FFFFD3AC 8, 16, 32         DATA_67       —       16       H'FFFFD3AE 8, 16   |        |          | LAFM1      | _            | 16             | H'FFFFD3A6 | 16             |
| DATA_45 — 16 H'FFFFD3AC 8, 16, 32  DATA_67 — 16 H'FFFFD3AE 8, 16  |        |          | DATA_01    | _            | 16             | H'FFFFD3A8 | 8, 16, 32      |
| DATA_67 — 16 H'FFFFD3AE 8, 16   |        |          | DATA_23    | _            | 16             | H'FFFFD3AA | 8, 16          |
|   |        |          | DATA_45    | _            | 16             | H'FFFFD3AC | 8, 16, 32      |
| CONTROL1 — 16 H'FFFFD3B0 8, 16  |        |          | DATA_67    | _            | 16             | H'FFFFD3AE | 8, 16          |
|   |        |          | CONTROL1   | _            | 16             | H'FFFFD3B0 | 8, 16          |

| Module         | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|----------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-          | MB[22].  | CONTROL0_H | _            | 16             | H'FFFFD3C0 | 16, 32         |
| TL1<br>(RCAN_  |          | CONTROL0_L | _            | 16             | H'FFFFD3C2 | 16             |
| (110AIN_<br>A) |          | LAFM0      | _            | 16             | H'FFFFD3C4 | 16, 32         |
|                |          | LAFM1      | _            | 16             | H'FFFFD3C6 | 16             |
|                |          | DATA_01    | _            | 16             | H'FFFFD3C8 | 8, 16, 32      |
|                |          | DATA_23    | _            | 16             | H'FFFFD3CA | 8, 16          |
|                |          | DATA_45    | _            | 16             | H'FFFFD3CC | 8, 16, 32      |
|                |          | DATA_67    | _            | 16             | H'FFFFD3CE | 8, 16          |
|                |          | CONTROL1   | _            | 16             | H'FFFFD3D0 | 8, 16          |
|                | MB[23].  | CONTROL0_H | _            | 16             | H'FFFFD3E0 | 16, 32         |
|                |          | CONTROL0_L | _            | 16             | H'FFFFD3E2 | 16             |
|                |          | LAFM0      | _            | 16             | H'FFFFD3E4 | 16, 32         |
|                |          | LAFM1      | _            | 16             | H'FFFFD3E6 | 16             |
|                |          | DATA_01    | _            | 16             | H'FFFFD3E8 | 8, 16, 32      |
|                |          | DATA_23    | _            | 16             | H'FFFFD3EA | 8, 16          |
|                |          | DATA_45    | _            | 16             | H'FFFFD3EC | 8, 16, 32      |
|                |          | DATA_67    | _            | 16             | H'FFFFD3EE | 8, 16          |
|                |          | CONTROL1   | _            | 16             | H'FFFFD3F0 | 8, 16          |
|                | MB[24].  | CONTROL0_H | _            | 16             | H'FFFFD400 | 16, 32         |
|                |          | CONTROL0_L | _            | 16             | H'FFFFD402 | 16             |
|                |          | LAFM0      | _            | 16             | H'FFFFD404 | 16, 32         |
|                |          | LAFM1      | _            | 16             | H'FFFFD406 | 16             |
|                |          | DATA_01    | _            | 16             | H'FFFFD408 | 8, 16, 32      |
|                |          | DATA_23    | _            | 16             | H'FFFFD40A | 8, 16          |
|                |          | DATA_45    | _            | 16             | H'FFFFD40C | 8, 16, 32      |
|                |          | DATA_67    | _            | 16             | H'FFFFD40E | 8, 16          |
|                |          | CONTROL1   | _            | 16             | H'FFFFD410 | 8, 16          |
|                |          | TTT        | _            | 16             | H'FFFFD414 | 16             |
|                |          | TTCONTROL  | _            | 16             | H'FFFFD416 | 16             |
|                | MB[25].  | CONTROL0_H | _            | 16             | H'FFFFD420 | 16, 32         |

| RCAN- TL1 (RCAN_ A)  MB[25]. CONTROLO_L — 16 H'FFFD422 16  LAFM0 — 16 H'FFFD424 16, 32  LAFM1 — 16 H'FFFD426 16  DATA_01 — 16 H'FFFD428 8, 16, 32  DATA_23 — 16 H'FFFD42A 8, 16  DATA_67 — 16 H'FFFD42E 8, 16  CONTROL1 — 16 H'FFFD430 8, 16  TTT — 16 H'FFFD434 16  TTCONTROL — 16 H'FFFD436 16  MB[26]. CONTROLO_H — 16 H'FFFD440 16, 32  CONTROLO_L — 16 H'FFFD442 16  LAFM0 — 16 H'FFFD444 16, 32  LAFM1 — 16 H'FFFD446 16  DATA_01 — 16 H'FFFD448 8, 16, 32  DATA_23 — 16 H'FFFD448 8, 16, 32  DATA_23 — 16 H'FFFD448 8, 16, 32  DATA_23 — 16 H'FFFD440 8, 16 |
|--|
| CAPMI  |
| A)  LAFM1 — 16 H'FFFD426 16  DATA_01 — 16 H'FFFD428 8, 16, 32  DATA_23 — 16 H'FFFD42A 8, 16  DATA_45 — 16 H'FFFD42C 8, 16, 32  DATA_67 — 16 H'FFFD42E 8, 16  CONTROL1 — 16 H'FFFD430 8, 16  TTT — 16 H'FFFD436 16  TTCONTROL — 16 H'FFFD436 16  MB[26]. CONTROL0_H — 16 H'FFFD440 16, 32  CONTROL0_L — 16 H'FFFD442 16  LAFM0 — 16 H'FFFD444 16, 32  LAFM1 — 16 H'FFFD446 16  DATA_01 — 16 H'FFFD448 8, 16, 32  DATA_23 — 16 H'FFFD44A 8, 16   |
| DATA_23 — 16 H'FFFD42A 8, 16  DATA_45 — 16 H'FFFD42C 8, 16, 32  DATA_67 — 16 H'FFFD42E 8, 16  CONTROL1 — 16 H'FFFD430 8, 16  TTT — 16 H'FFFD434 16  TTCONTROL — 16 H'FFFD436 16  MB[26]. CONTROL0_H — 16 H'FFFD440 16, 32  CONTROL0_L — 16 H'FFFD442 16  LAFM0 — 16 H'FFFD444 16, 32  LAFM1 — 16 H'FFFD446 16  DATA_01 — 16 H'FFFD448 8, 16, 32  DATA_23 — 16 H'FFFD44A 8, 16  |
| DATA_45 — 16 H'FFFD42C 8, 16, 32  DATA_67 — 16 H'FFFD42E 8, 16  CONTROL1 — 16 H'FFFD430 8, 16  TTT — 16 H'FFFD434 16  TTCONTROL — 16 H'FFFD436 16  MB[26]. CONTROL0_H — 16 H'FFFD440 16, 32  CONTROL0_L — 16 H'FFFD442 16  LAFM0 — 16 H'FFFD444 16, 32  LAFM1 — 16 H'FFFD446 16  DATA_01 — 16 H'FFFD448 8, 16, 32  DATA_23 — 16 H'FFFD44A 8, 16  |
| DATA_67 — 16 H'FFFD42E 8, 16  CONTROL1 — 16 H'FFFFD430 8, 16  TTT — 16 H'FFFFD434 16  TTCONTROL — 16 H'FFFFD436 16  MB[26]. CONTROL0_H — 16 H'FFFFD440 16, 32  CONTROL0_L — 16 H'FFFFD442 16  LAFM0 — 16 H'FFFFD444 16, 32  LAFM1 — 16 H'FFFFD446 16  DATA_01 — 16 H'FFFFD448 8, 16, 32  DATA_23 — 16 H'FFFFD44A 8, 16   |
| CONTROL1 — 16 H'FFFD430 8, 16  TTT — 16 H'FFFD434 16  TTCONTROL — 16 H'FFFD436 16  MB[26]. CONTROL0_H — 16 H'FFFD440 16, 32  CONTROL0_L — 16 H'FFFD442 16  LAFM0 — 16 H'FFFD444 16, 32  LAFM1 — 16 H'FFFD446 16  DATA_01 — 16 H'FFFD448 8, 16, 32  DATA_23 — 16 H'FFFD44A 8, 16  |
| TTT — 16 H'FFFD434 16  TTCONTROL — 16 H'FFFD436 16  MB[26]. CONTROLO_H — 16 H'FFFFD440 16, 32  CONTROLO_L — 16 H'FFFFD442 16  LAFM0 — 16 H'FFFFD444 16, 32  LAFM1 — 16 H'FFFFD446 16  DATA_01 — 16 H'FFFFD448 8, 16, 32  DATA_23 — 16 H'FFFFD44A 8, 16   |
| TTCONTROL — 16 H'FFFD436 16  MB[26]. CONTROLO_H — 16 H'FFFFD440 16, 32  CONTROLO_L — 16 H'FFFFD442 16  LAFM0 — 16 H'FFFFD444 16, 32  LAFM1 — 16 H'FFFFD446 16  DATA_01 — 16 H'FFFFD448 8, 16, 32  DATA_23 — 16 H'FFFFD44A 8, 16  |
| MB[26]. CONTROLO_H — 16 H'FFFFD440 16, 32  CONTROLO_L — 16 H'FFFFD442 16  LAFM0 — 16 H'FFFFD444 16, 32  LAFM1 — 16 H'FFFFD446 16  DATA_01 — 16 H'FFFFD448 8, 16, 32  DATA_23 — 16 H'FFFFD44A 8, 16   |
| CONTROLO_L — 16 H'FFFFD442 16  LAFM0 — 16 H'FFFFD444 16, 32  LAFM1 — 16 H'FFFFD446 16  DATA_01 — 16 H'FFFFD448 8, 16, 32  DATA_23 — 16 H'FFFFD44A 8, 16  |
| LAFM0 — 16 H'FFFFD444 16, 32  LAFM1 — 16 H'FFFFD446 16  DATA_01 — 16 H'FFFFD448 8, 16, 32  DATA_23 — 16 H'FFFFD44A 8, 16   |
| LAFM1       —       16       H'FFFFD446       16         DATA_01       —       16       H'FFFFD448       8, 16, 32         DATA_23       —       16       H'FFFFD44A       8, 16   |
| DATA_01         —         16         H'FFFFD448         8, 16, 32           DATA_23         —         16         H'FFFFD44A         8, 16  |
| DATA_23 — 16 H'FFFFD44A 8, 16  |
|  |
| DATA_45 — 16 H'FFFFD44C 8, 16, 32  |
|  |
| DATA_67 — 16 H'FFFFD44E 8, 16  |
| CONTROL1 — 16 H'FFFFD450 8, 16   |
| TTT — 16 H'FFFFD454 16   |
| TTCONTROL — 16 H'FFFFD456 16   |
| MB[27]. CONTROL0_H — 16 H'FFFFD460 16, 32  |
| CONTROL0_L — 16 H'FFFFD462 16  |
| LAFM0 — 16 H'FFFFD464 16, 32   |
| LAFM1 — 16 H'FFFFD466 16   |
| DATA_01 — 16 H'FFFFD468 8, 16, 32  |
| DATA_23 — 16 H'FFFFD46A 8, 16  |
| DATA_45 — 16 H'FFFFD46C 8, 16, 32  |
| DATA_67 — 16 H'FFFFD46E 8, 16  |
| CONTROL1 — 16 H'FFFFD470 8, 16   |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[27].  | TTT        | _            | 16             | H'FFFFD474 | 16             |
| TL1<br>(RCAN_ |          | TTCONTROL  | _            | 16             | H'FFFFD476 | 16             |
| A)            | MB[28].  | CONTROL0_H | _            | 16             | H'FFFFD480 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD482 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD484 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD486 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFD488 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFD48A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFD48C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD48E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD490 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFD494 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFD496 | 16             |
| •             | MB[29].  | CONTROL0_H | _            | 16             | H'FFFFD4A0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD4A2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD4A4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD4A6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFD4A8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFD4AA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFD4AC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFD4AE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFD4B0 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFD4B4 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFD4B6 | 16             |
| •             | MB[30].  | CONTROL0_H | _            | 16             | H'FFFFD4C0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFD4C2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFD4C4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFD4C6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFD4C8 | 8, 16, 32      |
|               |          | DATA_23    |              | 16             | H'FFFFD4CA | 8, 16          |

| Module        | Register                             | Name                   | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|--------------------------------------|------------------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[30].                              | DATA_45                | _            | 16             | H'FFFFD4CC | 8, 16, 32      |
| TL1<br>(RCAN_ |                                      | DATA_67                | _            | 16             | H'FFFFD4CE | 8, 16          |
| (110AN_<br>A) |                                      | CONTROL1               | _            | 16             | H'FFFFD4D0 | 8, 16          |
|               |                                      | TIMESTAMP              | _            | 16             | H'FFFFD4D2 | 16             |
|               |                                      | TTT                    | _            | 16             | H'FFFFD4D4 | 16             |
|               | MB[31].                              | CONTROL0_H             | _            | 16             | H'FFFFD4E0 | 16, 32         |
|               |                                      | CONTROL0_L             | _            | 16             | H'FFFFD4E2 | 16             |
|               |                                      | LAFM0                  | _            | 16             | H'FFFFD4E4 | 16, 32         |
|               |                                      | LAFM1                  | _            | 16             | H'FFFFD4E6 | 16             |
|               |                                      | DATA_01                | _            | 16             | H'FFFFD4E8 | 8, 16, 32      |
|               |                                      | DATA_23                | _            | 16             | H'FFFFD4EA | 8, 16          |
|               |                                      | DATA_45                | _            | 16             | H'FFFFD4EC | 8, 16, 32      |
|               |                                      | DATA_67                | _            | 16             | H'FFFFD4EE | 8, 16          |
|               |                                      | CONTROL1               | _            | 16             | H'FFFFD4F0 | 8, 16          |
|               |                                      | TIMESTAMP              | _            | 16             | H'FFFFD4F2 | 16             |
|               | Message buffer error status register |                        | MBESR        | 16             | H'FFFFD600 | 16             |
|               | Message<br>register                  | buffer error control   | MBECR        | 16             | H'FFFFD602 | 16             |
| RCAN-         | Master co                            | ontrol register        | MCR          | 16             | H'FFFFD800 | 16             |
| TL1<br>(RCAN_ | General status register              |                        | GSR          | 16             | H'FFFFD802 | 16             |
| (HCAN_<br>B)  | Bit configuration register 1         |                        | BCR1         | 16             | H'FFFFD804 | 16             |
|               | Bit configuration register 0         |                        | BCR0         | 16             | H'FFFFD806 | 16             |
|               | Interrupt                            | register               | IRR          | 16             | H'FFFFD808 | 16             |
|               | Interrupt                            | mask register          | IMR          | 16             | H'FFFFD80A | 16             |
|               | Error cou                            | nter register          | TEC/REC      | 16             | H'FFFFD80C | 16             |
|               | Transmit                             | pending register 1     | TXPR1        | 16             | H'FFFFD820 | 32             |
|               | Transmit                             | pending register 0     | TXPR0        | 16             | _          |                |
|               | Transmit                             | cancel register 1      | TXCR1        | 16             | H'FFFFD828 | 16             |
|               | Transmit                             | cancel register 0      | TXCR0        | 16             | H'FFFFD82A | 16             |
|               | Transmit                             | acknowledge register 1 | TXACK1       | 16             | H'FFFFD830 | 16             |
|               | Transmit                             | acknowledge register 0 | TXACK0       | 16             | H'FFFFD832 | 16             |

| Module        | Register                | Name                       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|-------------------------|----------------------------|--------------|----------------|------------|----------------|
| RCAN-         | Abort ack               | nowledge register 1        | ABACK1       | 16             | H'FFFFD838 | 16             |
| TL1<br>(RCAN_ | Abort ack               | nowledge register 0        | ABACK0       | 16             | H'FFFFD83A | 16             |
| B)            | Data fram<br>register 1 | ne receive pending         | RXPR1        | 16             | H'FFFFD840 | 16             |
|               | Data fram<br>register 0 | ne receive pending         | RXPR0        | 16             | H'FFFFD842 | 16             |
|               | Remote for register 1   | rame receive pending       | RFPR1        | 16             | H'FFFFD848 | 16             |
|               | Remote for register 0   | rame receive pending       | RFPR0        | 16             | H'FFFFD84A | 16             |
|               | Mailbox ir              | nterrupt mask register 1   | MBIMR1       | 16             | H'FFFFD850 | 16             |
|               | Mailbox ir              | nterrupt mask register 0   | MBIMR0       | 16             | H'FFFFD852 | 16             |
|               | Unread m<br>register 1  | nessage status             | UMSR1        | 16             | H'FFFFD858 | 16             |
|               | Unread m<br>register 0  | nessage status             | UMSR0        | 16             | H'FFFFD85A | 16             |
|               | Timer trig              | ger control register 0     | TTCR0        | 16             | H'FFFFD880 | 16             |
|               | Cycle ma<br>window re   | ximum/tx-enable<br>egister | CMAX_TEW     | 16             | H'FFFFD884 | 16             |
|               | Reference               | e trigger offset register  | RFTROFF      | 16             | H'FFFFD886 | 16             |
|               | Timer sta               | tus register               | TSR          | 16             | H'FFFFD888 | 16             |
|               | Cycle cou               | ınter register             | CCR          | 16             | H'FFFFD88A | 16             |
|               | Timer cou               | unter register             | TCNTR        | 16             | H'FFFFD88C | 16             |
|               | Cycle time              | e register                 | CYCTR        | 16             | H'FFFFD890 | 16             |
|               | Reference               | e mark register            | RFMK         | 16             | H'FFFFD894 | 16             |
|               | Timer cor               | mpare match register 0     | TCMR0        | 16             | H'FFFFD898 | 16             |
|               | Timer cor               | mpare match register 1     | TCMR1        | 16             | H'FFFFD89C | 16             |
|               | Timer cor               | mpare match register 2     | TCMR2        | 16             | H'FFFFD8A0 | 16             |
|               | Tx-trigger              | time selection register    | TTTSEL       | 16             | H'FFFFD8A4 | 16             |
|               | MB[0].                  | CONTROL0_H                 | _            | 16             | H'FFFFD900 | 16, 32         |
|               |                         | CONTROL0_L                 | _            | 16             | H'FFFFD902 | 16             |
|               |                         | LAFM0                      | _            | 16             | H'FFFFD904 | 16, 32         |
|               |                         | LAFM1                      | _            | 16             | H'FFFFD906 | 16             |

| RCAN-TL1 (RCAN_B)   | Module | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---|--------|----------|------------|--------------|----------------|------------|----------------|
| RCAN_B    DATA_45   | -      | MB[0].   | DATA_01    | _            | 16             | H'FFFFD908 | 8, 16, 32      |
| B)  |        |          | DATA_23    | _            | 16             | H'FFFFD90A | 8, 16          |
| CONTROL1 — 16 H'FFFFD910 8, 16  TIMESTAMP — 16 H'FFFFD912 16  MB[1]. CONTROLO_H — 16 H'FFFFD920 16, 32  CONTROLO_L — 16 H'FFFFD922 16  LAFM0 — 16 H'FFFFD924 16, 32  LAFM1 — 16 H'FFFFD926 16  DATA_01 — 16 H'FFFFD928 8, 16, 32  DATA_23 — 16 H'FFFFD92C 8, 16, 32  DATA_67 — 16 H'FFFFD92E 8, 16  CONTROL1 — 16 H'FFFFD92 8, 16  TIMESTAMP — 16 H'FFFFD93 16  MB[2]. CONTROLO_H — 16 H'FFFFD941 16, 32  LAFM1 — 16 H'FFFFD941 16, 32  LAFM0 — 16 H'FFFFD941 16, 32  LAFM1 — 16 H'FFFFD941 16, 32  LAFM1 — 16 H'FFFFD941 16, 32  DATA_23 — 16 H'FFFFD942 16  DATA_23 — 16 H'FFFFD94 16, 32  DATA_23 — 16 H'FFFFD94 16, 32  DATA_23 — 16 H'FFFFD94 8, 16  DATA_45 — 16 H'FFFFD94 8, 16  DATA_45 — 16 H'FFFFD94 8, 16  DATA_67 — 16 H'FFFFD95 8, 16  MB[3]. CONTROLO_H — 16 H'FFFFD95 16  MB[3]. CONTROLO_H — 16 H'FFFFD96 16, 32  CONTROLO_L — 16 H'FFFFD96 16, 32  CONTROLO_L — 16 H'FFFFD96 16, 32  |        |          | DATA_45    | _            | 16             | H'FFFFD90C | 8, 16, 32      |
| TIMESTAMP   |        |          | DATA_67    | _            | 16             | H'FFFFD90E | 8, 16          |
| MB[1]. CONTROLO_H — 16 H'FFFD920 16, 32  CONTROLO_L — 16 H'FFFD922 16  LAFM0 — 16 H'FFFD922 16  LAFM1 — 16 H'FFFD926 16  DATA_01 — 16 H'FFFD928 8, 16, 32  DATA_23 — 16 H'FFFD928 8, 16  DATA_45 — 16 H'FFFD922 8, 16  CONTROL1 — 16 H'FFFD922 8, 16  CONTROL1 — 16 H'FFFD930 16, 32  MB[2]. CONTROL0_H — 16 H'FFFD940 16, 32  CONTROL0_L — 16 H'FFFD940 16, 32  LAFM1 — 16 H'FFFD940 16, 32  LAFM1 — 16 H'FFFD940 16  DATA_23 — 16 H'FFFD940 16  DATA_23 — 16 H'FFFD940 16  DATA_01 — 16 H'FFFD940 16  DATA_01 — 16 H'FFFD940 16  DATA_01 — 16 H'FFFD940 16  DATA_23 — 16 H'FFFD940 16  DATA_45 — 16 H'FFFD940 8, 16, 32  DATA_67 — 16 H'FFFD940 8, 16  CONTROL1 — 16 H'FFFD940 8, 16  TIMESTAMP — 16 H'FFFD950 8, 16  CONTROL1 — 16 H'FFFD950 16  MB[3]. CONTROLO_H — 16 H'FFFD960 16, 32  CONTROLO_L — 16 H'FFFD960 16, 32  CONTROLO_L — 16 H'FFFD960 16, 32  CONTROLO_L — 16 H'FFFD960 16, 32   |        |          | CONTROL1   | _            | 16             | H'FFFFD910 | 8, 16          |
| CONTROLO_L  LAFM0 — 16 H'FFFFD922 16  LAFM1 — 16 H'FFFFD924 16, 32  LAFM1 — 16 H'FFFFD926 16  DATA_01 — 16 H'FFFFD928 8, 16, 32  DATA_23 — 16 H'FFFD928 8, 16  DATA_45 — 16 H'FFFD922 8, 16  CONTROL1 — 16 H'FFFD922 8, 16  CONTROL1 — 16 H'FFFD930 8, 16  TIMESTAMP — 16 H'FFFD932 16  MB[2]. CONTROLO_H — 16 H'FFFD942 16  LAFM0 — 16 H'FFFD942 16  DATA_01 — 16 H'FFFD944 16, 32  LAFM1 — 16 H'FFFD946 16  DATA_23 — 16 H'FFFD948 8, 16, 32  DATA_23 — 16 H'FFFD948 8, 16  DATA_45 — 16 H'FFFD948 8, 16  CONTROL1 — 16 H'FFFD948 8, 16  DATA_67 — 16 H'FFFD946 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD952 16  MB[3]. CONTROLO_H — 16 H'FFFD962 16  LAFM0 — 16 H'FFFFD962 16  LAFM0 — 16 H'FFFFD962 16  |        |          | TIMESTAMP  | _            | 16             | H'FFFFD912 | 16             |
| LAFM0 — 16 H'FFFFD924 16, 32  LAFM1 — 16 H'FFFFD926 16  DATA_01 — 16 H'FFFFD928 8, 16, 32  DATA_23 — 16 H'FFFFD92 8, 16, 32  DATA_45 — 16 H'FFFFD92 8, 16  CONTROL1 — 16 H'FFFFD92 8, 16  CONTROL1 — 16 H'FFFFD93 8, 16  TIMESTAMP — 16 H'FFFFD93 16  MB[2]. CONTROL0_H — 16 H'FFFFD94 16, 32  CONTROL0_L — 16 H'FFFFD94 16, 32  LAFM1 — 16 H'FFFFD94 16, 32  LAFM1 — 16 H'FFFFD94 8, 16  DATA_23 — 16 H'FFFFD94 8, 16  DATA_23 — 16 H'FFFFD94 8, 16  DATA_67 — 16 H'FFFFD94 8, 16  CONTROL1 — 16 H'FFFFD94 8, 16  CONTROL1 — 16 H'FFFFD95 8, 16  CONTROL1 — 16 H'FFFFD95 8, 16  CONTROL1 — 16 H'FFFFD95 16  MB[3]. CONTROL0_H — 16 H'FFFFD96 16, 32  CONTROL0_L — 16 H'FFFFD96 16, 32  CONTROL0_L — 16 H'FFFFD96 16, 32  |        | MB[1].   | CONTROL0_H | _            | 16             | H'FFFFD920 | 16, 32         |
| LAFM1 — 16 H'FFFD926 16  DATA_01 — 16 H'FFFD928 8, 16, 32  DATA_23 — 16 H'FFFD92A 8, 16  DATA_45 — 16 H'FFFD92C 8, 16, 32  DATA_67 — 16 H'FFFD92E 8, 16  CONTROL1 — 16 H'FFFD932 16  MB[2]. CONTROL0_H — 16 H'FFFD942 16  LAFM0 — 16 H'FFFD944 16, 32  LAFM1 — 16 H'FFFD946 16  DATA_01 — 16 H'FFFD948 8, 16  DATA_23 — 16 H'FFFD948 8, 16  DATA_45 — 16 H'FFFD948 8, 16  CONTROL1 — 16 H'FFFD948 8, 16  DATA_67 — 16 H'FFFD94C 8, 16, 32  DATA_67 — 16 H'FFFD94E 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD952 16  MB[3]. CONTROL0_H — 16 H'FFFD962 16  LAFM0 — 16 H'FFFD962 16   |        |          | CONTROL0_L | _            | 16             | H'FFFFD922 | 16             |
| DATA_01         —         16         H'FFFD928         8, 16, 32           DATA_23         —         16         H'FFFD92A         8, 16           DATA_45         —         16         H'FFFD92C         8, 16, 32           DATA_67         —         16         H'FFFD92E         8, 16           CONTROL1         —         16         H'FFFD930         8, 16           TIMESTAMP         —         16         H'FFFD932         16           MB[2].         CONTROLO_H         —         16         H'FFFD940         16, 32           CONTROLO_L         —         16         H'FFFFD942         16           LAFM0         —         16         H'FFFFD944         16, 32           LAFM1         —         16         H'FFFFD946         16           DATA_01         —         16         H'FFFFD948         8, 16, 32           DATA_23         —         16         H'FFFFD94C         8, 16, 32           DATA_67         —         16         H'FFFFD94E         8, 16           CONTROL1         —         16         H'FFFFD950         8, 16           TIMESTAMP         —         16         H'FFFFD960         16, 32 |        |          | LAFM0      | _            | 16             | H'FFFFD924 | 16, 32         |
| DATA_23 — 16 H'FFFFD92A 8, 16  DATA_45 — 16 H'FFFFD92C 8, 16, 32  DATA_67 — 16 H'FFFFD92E 8, 16  CONTROL1 — 16 H'FFFFD930 8, 16  TIMESTAMP — 16 H'FFFFD932 16  MB[2]. CONTROL0_H — 16 H'FFFFD940 16, 32  CONTROL0_L — 16 H'FFFFD942 16  LAFM0 — 16 H'FFFFD944 16, 32  LAFM1 — 16 H'FFFFD946 16  DATA_01 — 16 H'FFFFD948 8, 16, 32  DATA_23 — 16 H'FFFFD94A 8, 16  DATA_45 — 16 H'FFFFD94C 8, 16, 32  DATA_67 — 16 H'FFFFD94E 8, 16  CONTROL1 — 16 H'FFFFD950 8, 16  TIMESTAMP — 16 H'FFFFD952 16  MB[3]. CONTROL0_H — 16 H'FFFFD962 16  LAFM0 — 16 H'FFFFD962 16  |        |          | LAFM1      | _            | 16             | H'FFFFD926 | 16             |
| DATA_45 — 16 H'FFFD92C 8, 16, 32  DATA_67 — 16 H'FFFD92E 8, 16  CONTROL1 — 16 H'FFFD930 8, 16  TIMESTAMP — 16 H'FFFD932 16  MB[2]. CONTROL0_H — 16 H'FFFD940 16, 32  CONTROL0_L — 16 H'FFFD942 16  LAFM0 — 16 H'FFFD946 16  DATA_01 — 16 H'FFFD948 8, 16, 32  DATA_23 — 16 H'FFFD94A 8, 16  DATA_45 — 16 H'FFFD94C 8, 16, 32  DATA_67 — 16 H'FFFD94E 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD952 16  MB[3]. CONTROL0_H — 16 H'FFFD960 16, 32  CONTROL0_L — 16 H'FFFD962 16  LAFM0 — 16 H'FFFD962 16  |        |          | DATA_01    | _            | 16             | H'FFFFD928 | 8, 16, 32      |
| DATA_67 — 16 H'FFFD92E 8, 16  CONTROL1 — 16 H'FFFD930 8, 16  TIMESTAMP — 16 H'FFFD932 16  MB[2]. CONTROL0_H — 16 H'FFFD942 16  LAFM0 — 16 H'FFFD942 16  LAFM1 — 16 H'FFFD946 16  DATA_01 — 16 H'FFFD948 8, 16, 32  DATA_23 — 16 H'FFFD94A 8, 16  DATA_45 — 16 H'FFFD94C 8, 16, 32  DATA_67 — 16 H'FFFD94E 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD950 16, 32  CONTROL0_H — 16 H'FFFD960 16, 32  CONTROL0_L — 16 H'FFFD962 16  MB[3]. CONTROL0_H — 16 H'FFFFD962 16  LAFM0 — 16 H'FFFFD964 16, 32   |        |          | DATA_23    | _            | 16             | H'FFFFD92A | 8, 16          |
| CONTROL1 — 16 H'FFFD930 8, 16  TIMESTAMP — 16 H'FFFD932 16  MB[2]. CONTROL0_H — 16 H'FFFD940 16, 32  CONTROL0_L — 16 H'FFFD942 16  LAFM0 — 16 H'FFFD944 16, 32  LAFM1 — 16 H'FFFD946 16  DATA_01 — 16 H'FFFD948 8, 16, 32  DATA_23 — 16 H'FFFD94A 8, 16  DATA_45 — 16 H'FFFD94C 8, 16, 32  DATA_67 — 16 H'FFFD94E 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD950 16, 32  CONTROL0_H — 16 H'FFFD960 16, 32  CONTROL0_L — 16 H'FFFD962 16  MB[3]. CONTROL0_L — 16 H'FFFD962 16  LAFM0 — 16 H'FFFD964 16, 32   |        |          | DATA_45    | _            | 16             | H'FFFFD92C | 8, 16, 32      |
| TIMESTAMP — 16 H'FFFD932 16  MB[2]. CONTROLO_H — 16 H'FFFFD940 16, 32  CONTROLO_L — 16 H'FFFFD942 16  LAFM0 — 16 H'FFFFD944 16, 32  LAFM1 — 16 H'FFFFD946 16  DATA_01 — 16 H'FFFFD948 8, 16, 32  DATA_23 — 16 H'FFFFD94A 8, 16  DATA_45 — 16 H'FFFFD94C 8, 16, 32  DATA_67 — 16 H'FFFFD94E 8, 16  CONTROL1 — 16 H'FFFFD950 8, 16  TIMESTAMP — 16 H'FFFFD952 16  MB[3]. CONTROLO_H — 16 H'FFFFD962 16  LAFM0 — 16 H'FFFFD962 16  |        |          | DATA_67    | _            | 16             | H'FFFFD92E | 8, 16          |
| MB[2]. CONTROLO_H — 16 H'FFFD940 16, 32  CONTROLO_L — 16 H'FFFD942 16  LAFM0 — 16 H'FFFD944 16, 32  LAFM1 — 16 H'FFFD946 16  DATA_01 — 16 H'FFFD948 8, 16, 32  DATA_23 — 16 H'FFFD94A 8, 16  DATA_45 — 16 H'FFFD94C 8, 16, 32  DATA_67 — 16 H'FFFD94E 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD952 16  MB[3]. CONTROLO_H — 16 H'FFFD962 16  LAFM0 — 16 H'FFFD964 16, 32   |        |          | CONTROL1   | _            | 16             | H'FFFFD930 | 8, 16          |
| CONTROLO_L — 16 H'FFFD942 16  LAFM0 — 16 H'FFFD944 16, 32  LAFM1 — 16 H'FFFD946 16  DATA_01 — 16 H'FFFD948 8, 16, 32  DATA_23 — 16 H'FFFD94A 8, 16  DATA_45 — 16 H'FFFD94C 8, 16, 32  DATA_67 — 16 H'FFFD94E 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD952 16  MB[3]. CONTROLO_H — 16 H'FFFD962 16  LAFM0 — 16 H'FFFD964 16, 32  |        |          | TIMESTAMP  | _            | 16             | H'FFFFD932 | 16             |
| LAFM0 — 16 H'FFFD944 16, 32  LAFM1 — 16 H'FFFD946 16  DATA_01 — 16 H'FFFD948 8, 16, 32  DATA_23 — 16 H'FFFD94C 8, 16, 32  DATA_45 — 16 H'FFFD94C 8, 16, 32  DATA_67 — 16 H'FFFD94E 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD952 16  MB[3]. CONTROL0_H — 16 H'FFFD962 16  LAFM0 — 16 H'FFFD964 16, 32  |        | MB[2].   | CONTROL0_H | _            | 16             | H'FFFFD940 | 16, 32         |
| LAFM1 — 16 H'FFFD946 16  DATA_01 — 16 H'FFFD948 8, 16, 32  DATA_23 — 16 H'FFFD94A 8, 16  DATA_45 — 16 H'FFFD94C 8, 16, 32  DATA_67 — 16 H'FFFD94E 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD952 16  MB[3]. CONTROL0_H — 16 H'FFFD962 16  LAFM0 — 16 H'FFFD964 16, 32   |        |          | CONTROL0_L | _            | 16             | H'FFFFD942 | 16             |
| DATA_01       —       16       H'FFFFD948       8, 16, 32         DATA_23       —       16       H'FFFFD94A       8, 16         DATA_45       —       16       H'FFFFD94C       8, 16, 32         DATA_67       —       16       H'FFFFD94E       8, 16         CONTROL1       —       16       H'FFFFD950       8, 16         TIMESTAMP       —       16       H'FFFFD952       16         MB[3].       CONTROLO_H       —       16       H'FFFFD960       16, 32         CONTROLO_L       —       16       H'FFFFD964       16, 32  |        |          | LAFM0      | _            | 16             | H'FFFFD944 | 16, 32         |
| DATA_23 — 16 H'FFFD94A 8, 16  DATA_45 — 16 H'FFFD94C 8, 16, 32  DATA_67 — 16 H'FFFD94E 8, 16  CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD952 16  MB[3]. CONTROL0_H — 16 H'FFFD960 16, 32  CONTROL0_L — 16 H'FFFD962 16  LAFMO — 16 H'FFFD964 16, 32  |        |          | LAFM1      | _            | 16             | H'FFFFD946 | 16             |
| DATA_45       —       16       H'FFFFD94C       8, 16, 32         DATA_67       —       16       H'FFFFD94E       8, 16         CONTROL1       —       16       H'FFFFD950       8, 16         TIMESTAMP       —       16       H'FFFFD952       16         MB[3].       CONTROLO_H       —       16       H'FFFFD960       16, 32         CONTROLO_L       —       16       H'FFFFD962       16         LAFM0       —       16       H'FFFFD964       16, 32   |        |          | DATA_01    | _            | 16             | H'FFFFD948 | 8, 16, 32      |
| DATA_67       —       16       H'FFFFD94E       8, 16         CONTROL1       —       16       H'FFFFD950       8, 16         TIMESTAMP       —       16       H'FFFFD952       16         MB[3].       CONTROLO_H       —       16       H'FFFFD960       16, 32         CONTROLO_L       —       16       H'FFFFD962       16         LAFM0       —       16       H'FFFFD964       16, 32   |        |          | DATA_23    | _            | 16             | H'FFFFD94A | 8, 16          |
| CONTROL1 — 16 H'FFFD950 8, 16  TIMESTAMP — 16 H'FFFD952 16  MB[3]. CONTROL0_H — 16 H'FFFD960 16, 32  CONTROL0_L — 16 H'FFFD962 16  LAFM0 — 16 H'FFFD964 16, 32  |        |          | DATA_45    | _            | 16             | H'FFFFD94C | 8, 16, 32      |
| TIMESTAMP         —         16         H'FFFFD952         16           MB[3].         CONTROLO_H         —         16         H'FFFFD960         16, 32           CONTROLO_L         —         16         H'FFFFD962         16           LAFM0         —         16         H'FFFFD964         16, 32  |        |          | DATA_67    | _            | 16             | H'FFFFD94E | 8, 16          |
| MB[3]. CONTROLO_H — 16 H'FFFFD960 16, 32  CONTROLO_L — 16 H'FFFFD962 16  LAFM0 — 16 H'FFFFD964 16, 32   |        |          | CONTROL1   | _            | 16             | H'FFFFD950 | 8, 16          |
| CONTROLO_L — 16 H'FFFFD962 16  LAFM0 — 16 H'FFFFD964 16, 32   |        |          | TIMESTAMP  | _            | 16             | H'FFFFD952 | 16             |
| LAFM0 — 16 H'FFFFD964 16, 32  |        | MB[3].   | CONTROL0_H | _            | 16             | H'FFFFD960 | 16, 32         |
|   |        |          | CONTROL0_L | _            | 16             | H'FFFFD962 | 16             |
| LAFM1 — 16 H'FFFFD966 16  |        |          | LAFM0      | _            | 16             | H'FFFFD964 | 16, 32         |
|   |        |          | LAFM1      |              | 16             | H'FFFFD966 | 16             |

| RCAN- MB[3].       DATA_01       —       16       H'FFFD968       8, 16, 16, 16, 16         TL1 (RCAN_ (RCAN_ B))       DATA_23       —       16       H'FFFD96C       8, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16 |
|--|
| (RCAN_B) DATA_23 — 16 H*FFFD96A 8, 16 DATA_45 — 16 H*FFFD96C 8, 16, DATA_67 — 16 H*FFFD96E 8, 16   |
| DATA_45 — 16 H'FFFFD96C 8, 16, DATA_67 — 16 H'FFFFD96E 8, 16   |
| <u></u>  |
| CONTROL1 — 16 H'FFFFD970 8, 16   |
|  |
| TIMESTAMP — 16 H'FFFFD972 16   |
| MB[4]. CONTROL0_H — 16 H'FFFFD980 16, 32   |
| CONTROL0_L 16 H'FFFFD982 16  |
| LAFM0 — 16 H'FFFFD984 16, 32   |
| LAFM1 — 16 H'FFFFD986 16   |
| DATA_01 — 16 H'FFFFD988 8, 16,   |
| DATA_23 — 16 H'FFFFD98A 8, 16  |
| DATA_45 — 16 H'FFFFD98C 8, 16,   |
| DATA_67 — 16 H'FFFFD98E 8, 16  |
| CONTROL1 — 16 H'FFFFD990 8, 16   |
| TIMESTAMP — 16 H'FFFFD992 16   |
| MB[5]. CONTROLO_H 16 H'FFFFD9A0 16, 32   |
| CONTROLO_L — 16 H'FFFFD9A2 16  |
| LAFM0 — 16 H'FFFFD9A4 16, 32   |
| LAFM1 — 16 H'FFFFD9A6 16   |
| DATA_01 — 16 H'FFFFD9A8 8, 16,   |
| DATA_23 — 16 H'FFFFD9AA 8, 16  |
| DATA_45 — 16 H'FFFFD9AC 8, 16,   |
| DATA_67 — 16 H'FFFFD9AE 8, 16  |
| CONTROL1 — 16 H'FFFFD9B0 8, 16   |
| TIMESTAMP — 16 H'FFFFD9B2 16   |
| MB[6]. CONTROL0_H — 16 H'FFFFD9C0 16, 32   |
| CONTROLO_L — 16 H'FFFFD9C2 16  |
| LAFM0 — 16 H'FFFFD9C4 16, 32   |
| LAFM1 — 16 H'FFFFD9C6 16   |

| RCAN- MB[6].       DATA_01       —       16       H'FFFD9C8 8         TL1 (RCAN_ (RCAN_ B))       DATA_23       —       16       H'FFFFD9CC 8         DATA_45       —       16       H'FFFFD9CC 8         DATA_67       —       16       H'FFFFD9CE 8 | 8, 16<br>8, 16, 32<br>8, 16       |
|---|-----------------------------------|
| (RCAN_ B) DATA_45 — 16 H'FFFD9CC 8  | 8, 16, 32<br>8, 16<br>8, 16<br>16 |
| B) DATA_45 — 16 H'FFFD9CC 8   | 8, 16<br>8, 16<br>16              |
| DATA 67 — 16 H'FFFFD9CF 8   | 8, 16<br>16                       |
|   | 16                                |
| CONTROL1 — 16 H'FFFFD9D0 8  |                                   |
| TIMESTAMP — 16 H'FFFFD9D2   | 16 32                             |
| MB[7]. CONTROLO_H — 16 H'FFFFD9E0   | 10, 52                            |
| CONTROLO_L — 16 H'FFFFD9E2  | 16                                |
| LAFMO — 16 H'FFFD9E4  | 16, 32                            |
| LAFM1 — 16 H'FFFD9E6  | 16                                |
| DATA_01 — 16 H'FFFFD9E8 8   | 8, 16, 32                         |
| DATA_23 — 16 H'FFFFD9EA 8   | 8, 16                             |
| DATA_45 — 16 H'FFFFD9EC 8   | 8, 16, 32                         |
| DATA_67 — 16 H'FFFFD9EE 8   | 8, 16                             |
| CONTROL1 — 16 H'FFFFD9F0 8  | 8, 16                             |
| TIMESTAMP — 16 H'FFFFD9F2   | 16                                |
| MB[8]. CONTROLO_H — 16 H'FFFFDA00 1   | 16, 32                            |
| CONTROLO_L — 16 H'FFFFDA02 1  | 16                                |
| LAFMO — 16 H'FFFDA04  | 16, 32                            |
| LAFM1 — 16 H'FFFDA06  | 16                                |
| DATA_01 — 16 H'FFFFDA08 8   | 8, 16, 32                         |
| DATA_23 — 16 H'FFFFDA0A 8   | 8, 16                             |
| DATA_45 — 16 H'FFFDA0C 8  | 8, 16, 32                         |
| DATA_67 — 16 H'FFFFDA0E 8   | 8, 16                             |
| CONTROL1 — 16 H'FFFFDA10 8  | 8, 16                             |
| TIMESTAMP — 16 H'FFFDA12  | 16                                |
| MB[9]. CONTROLO_H — 16 H'FFFFDA20 1   | 16, 32                            |
| CONTROLO_L — 16 H'FFFFDA22 1  | 16                                |
| LAFM0 — 16 H'FFFDA24  | 16, 32                            |
| LAFM1 — 16 H'FFFDA26 1  | 16                                |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[9].   | DATA_01    | _            | 16             | H'FFFFDA28 | 8, 16, 32      |
| TL1<br>(RCAN_ |          | DATA_23    | _            | 16             | H'FFFFDA2A | 8, 16          |
| B)            |          | DATA_45    | _            | 16             | H'FFFFDA2C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDA2E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDA30 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFDA32 | 16             |
|               | MB[10].  | CONTROL0_H | _            | 16             | H'FFFFDA40 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDA42 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDA44 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDA46 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDA48 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDA4A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDA4C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDA4E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDA50 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFDA52 | 16             |
|               | MB[11].  | CONTROL0_H | _            | 16             | H'FFFFDA60 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDA62 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDA64 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDA66 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDA68 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDA6A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDA6C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDA6E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDA70 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFDA72 | 16             |
|               | MB[12].  | CONTROL0_H | _            | 16             | H'FFFFDA80 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDA82 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDA84 | 16, 32         |
|               |          | LAFM1      |              | 16             | H'FFFFDA86 | 16             |

| RCAN-TLI (RCAN_B)  | Module | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--|--------|----------|------------|--------------|----------------|------------|----------------|
| RCAN_B    DATA_45  | _      | MB[12].  | DATA_01    | _            | 16             | H'FFFFDA88 | 8, 16, 32      |
| B) DATA_45 — 16 H'FFFDA8C 8, 16, 32 DATA_67 — 16 H'FFFDA8E 8, 16 CONTROL1 — 16 H'FFFDA90 8, 16 TIMESTAMP — 16 H'FFFDA92 16  MB[13]. CONTROLO_H — 16 H'FFFDA42 16 LAFMO — 16 H'FFFDA44 16, 32 LAFM1 — 16 H'FFFDA46 16 DATA_01 — 16 H'FFFDAA8 8, 16, 32 DATA_23 — 16 H'FFFDAAC 8, 16, 32 DATA_67 — 16 H'FFFDABE 8, 16 CONTROLO_H — 16 H'FFFDABE 8, 16 CONTROLO_H — 16 H'FFFDACE 16 TIMESTAMP — 16 H'FFFDABE 16  MB[14]. CONTROLO_H — 16 H'FFFDACE 16, 32 CONTROLO_L — 16 H'FFFDACE 16, 32 LAFM1 — 16 H'FFFDACE 16 DATA_01 — 16 H'FFFDACE 16 DATA_01 — 16 H'FFFDACE 16  MB[14]. CONTROLO_H — 16 H'FFFDACE 16 LAFMO — 16 H'FFFDACE 16 DATA_01 — 16 H'FFFDACE 8, 16, 32 DATA_23 — 16 H'FFFDACE 16 DATA_01 — 16 H'FFFDACE 8, 16, 32 DATA_67 — 16 H'FFFDACE 8, 16, 32 DATA_67 — 16 H'FFFDACE 8, 16 CONTROL1 — 16 H'FFFDACE 8, 16 DATA_67 — 16 H'FFFDACE 8, 16 CONTROL1 — 16 H'FFFDACE 16 TIMESTAMP — 16 H'FFFDACE 8, 16 CONTROL1 — 16 H'FFFDACE 16 DATA_67 — 16 H'FFFDACE 8, 16 CONTROL0_H — 16 H'FFFDACE 16 TIMESTAMP — 16 H'FFFDACE 16  MB[15]. CONTROLO_H — 16 H'FFFDACE 16 LAFMO — 16 H'FFFDACE 16 DATA_01 — 16 H'FFFDACE 16 DA |        |          | DATA_23    | _            | 16             | H'FFFFDA8A | 8, 16          |
| CONTROL1   |        |          | DATA_45    | _            | 16             | H'FFFFDA8C | 8, 16, 32      |
| TIMESTAMP  |        |          | DATA_67    | _            | 16             | H'FFFFDA8E | 8, 16          |
| MB[13]. CONTROLO_H — 16 H'FFFDAA0 16, 32  CONTROLO_L — 16 H'FFFDAA2 16  LAFM0 — 16 H'FFFDAA4 16, 32  LAFM1 — 16 H'FFFDAA6 16  DATA_01 — 16 H'FFFDAA8 8, 16, 32  DATA_23 — 16 H'FFFDAAA 8, 16  DATA_45 — 16 H'FFFDAAE 8, 16  CONTROL1 — 16 H'FFFDAB2 16  MB[14]. CONTROLO_H — 16 H'FFFDAC0 16, 32  CONTROL0_L — 16 H'FFFDAC0 16, 32  LAFM1 — 16 H'FFFDAC0 16, 32  LAFM1 — 16 H'FFFDAC0 16, 32  DATA_23 — 16 H'FFFDAC0 16, 32  CONTROLO_L — 16 H'FFFDAC0 16, 32  DATA_01 — 16 H'FFFDAC0 16, 32  DATA_01 — 16 H'FFFDAC0 16  CONTROL1 — 16 H'FFFDAC0 16  TIMESTAMP — 16 H'FFFDAC0 16  MB[15]. CONTROLO_H — 16 H'FFFFDAC0 16, 32  CONTROLO_L — 16 H'FFFFDAC0 16, 32  CONTROLO_L — 16 H'FFFFDAC0 16, 32   |        |          | CONTROL1   | _            | 16             | H'FFFFDA90 | 8, 16          |
| CONTROLO_L  LAFM0 — 16 H'FFFDAA2 16  LAFM1 — 16 H'FFFDAA6 16  DATA_01 — 16 H'FFFDAA8 8, 16, 32  DATA_23 — 16 H'FFFDAA8 8, 16  DATA_45 — 16 H'FFFDAA2 8, 16  CONTROL1 — 16 H'FFFDAB0 8, 16  TIMESTAMP — 16 H'FFFDAB2 16  CONTROL0_L — 16 H'FFFDAC2 16  LAFM0 — 16 H'FFFDAC4 16, 32  LAFM1 — 16 H'FFFDAC6 16  DATA_23 — 16 H'FFFDAC8 8, 16, 32  DATA_67 — 16 H'FFFDAC2 16  CONTROL0_L — 16 H'FFFDAC2 16  DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDAC8 8, 16, 32  DATA_45 — 16 H'FFFDAC8 8, 16, 32  DATA_67 — 16 H'FFFDAC8 8, 16  CONTROL1 — 16 H'FFFDAC8 8, 16  TIMESTAMP — 16 H'FFFDAC8 8, 16  CONTROL1 — 16 H'FFFDAC8 8, 16  CONTROL0_L — 16 H'FFFDAC8 16, 32  CONTROL0_L — 16 H'FFFDAC8 16, 32  CONTROL0_L — 16 H'FFFFDAE2 16  LAFM0 — 16 H'FFFFDAE2 16  |        |          | TIMESTAMP  | _            | 16             | H'FFFFDA92 | 16             |
| LAFMO — 16 H'FFFDAA4 16, 32  LAFM1 — 16 H'FFFDAA6 16  DATA_01 — 16 H'FFFDAA8 8, 16, 32  DATA_23 — 16 H'FFFDAAA 8, 16  DATA_45 — 16 H'FFFDAAC 8, 16, 32  DATA_67 — 16 H'FFFDAAE 8, 16  CONTROL1 — 16 H'FFFDAB2 16  MB[14]. CONTROL0_H — 16 H'FFFDAC2 16  LAFMO — 16 H'FFFDAC4 16, 32  LAFM1 — 16 H'FFFDAC6 16  DATA_01 — 16 H'FFFDAC8 8, 16  DATA_23 — 16 H'FFFDAC8 8, 16  DATA_01 — 16 H'FFFDAC8 8, 16  DATA_01 — 16 H'FFFDAC8 8, 16  DATA_67 — 16 H'FFFDAC8 8, 16  CONTROL1 — 16 H'FFFDAC8 8, 16  DATA_67 — 16 H'FFFDAC8 8, 16  CONTROL1 — 16 H'FFFFDAC8 8, 16  CONTROL0_H — 16 H'FFFFDAC8 16  MB[15]. CONTROL0_H — 16 H'FFFFDAC9 16, 32  CONTROL0_L — 16 H'FFFFDAE2 16  LAFMO — 16 H'FFFFDAE2 16   |        | MB[13].  | CONTROL0_H | _            | 16             | H'FFFFDAA0 | 16, 32         |
| LAFM1 — 16 H'FFFDAA6 16  DATA_01 — 16 H'FFFDAA8 8, 16, 32  DATA_23 — 16 H'FFFDAAA 8, 16  DATA_45 — 16 H'FFFDAAC 8, 16, 32  DATA_67 — 16 H'FFFDAAE 8, 16  CONTROL1 — 16 H'FFFDAB2 16  MB[14]. CONTROL0_H — 16 H'FFFDAC2 16, 32  CONTROL0_L — 16 H'FFFDAC6 16, 32  LAFM1 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDAC8 8, 16, 32  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDACE 16  MB[15]. CONTROL0_H — 16 H'FFFDAE2 16  MB[15]. CONTROL0_H — 16 H'FFFDAE2 16  LAFM0 — 16 H'FFFDAE2 16   |        |          | CONTROL0_L | _            | 16             | H'FFFFDAA2 | 16             |
| DATA_01 — 16 H'FFFDAA8 8, 16, 32  DATA_23 — 16 H'FFFDAAA 8, 16  DATA_45 — 16 H'FFFDAAC 8, 16, 32  DATA_67 — 16 H'FFFDAAE 8, 16  CONTROL1 — 16 H'FFFDAB2 16  TIMESTAMP — 16 H'FFFDAC2 16, 32  CONTROL0_H — 16 H'FFFDAC2 16  LAFM0 — 16 H'FFFDAC6 16, 32  LAFM1 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDAC8 8, 16, 32  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDACE 8, 16  TIMESTAMP — 16 H'FFFDACE 16  MB[15]. CONTROL0_H — 16 H'FFFDAE2 16  MB[15]. CONTROL0_H — 16 H'FFFDAE2 16  CONTROL0_L — 16 H'FFFDAE2 16  MB[15]. CONTROL0_H — 16 H'FFFDAE2 16  LAFM0 — 16 H'FFFDAE2 16  |        |          | LAFM0      | _            | 16             | H'FFFFDAA4 | 16, 32         |
| DATA_23         —         16         H'FFFFDAAA 8, 16           DATA_45         —         16         H'FFFFDAAC 8, 16, 32           DATA_67         —         16         H'FFFFDAAE 8, 16           CONTROL1         —         16         H'FFFFDABO 8, 16           TIMESTAMP         —         16         H'FFFFDAB2 16           MB[14].         CONTROLO_H         —         16         H'FFFFDACO 16, 32           CONTROLO_L         —         16         H'FFFFDAC2 16           LAFM0         —         16         H'FFFFDAC4 16, 32           LAFM1         —         16         H'FFFFDAC8 8, 16, 32           DATA_01         —         16         H'FFFFDAC8 8, 16, 32           DATA_23         —         16         H'FFFFDACC 8, 16, 32           DATA_45         —         16         H'FFFFDACE 8, 16           DATA_67         —         16         H'FFFFDAC8 8, 16           CONTROL1         —         16         H'FFFFDAD2 16           MB[15].         CONTROLO_H         —         16         H'FFFFDAE2 16           LAFMO         —         16         H'FFFFDAE2 16         H  |        |          | LAFM1      | _            | 16             | H'FFFFDAA6 | 16             |
| DATA_45 — 16 H'FFFDAAC 8, 16, 32 DATA_67 — 16 H'FFFDAAE 8, 16  CONTROL1 — 16 H'FFFDABO 8, 16  TIMESTAMP — 16 H'FFFDACO 16, 32  CONTROLO_H — 16 H'FFFDAC2 16  LAFMO — 16 H'FFFDAC4 16, 32  LAFM1 — 16 H'FFFDAC8 8, 16, 32  DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_45 — 16 H'FFFDACA 8, 16  DATA_45 — 16 H'FFFDACE 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDAC 8, 16  TIMESTAMP — 16 H'FFFDAC2 16  MB[15]. CONTROLO_H — 16 H'FFFDAE2 16  MB[15]. CONTROLO_H — 16 H'FFFDAE2 16  CONTROLO_L — 16 H'FFFDAE2 16  LAFMO — 16 H'FFFDAE2 16   |        |          | DATA_01    | _            | 16             | H'FFFFDAA8 | 8, 16, 32      |
| DATA_67 — 16 H'FFFDAAE 8, 16  CONTROL1 — 16 H'FFFDABO 8, 16  TIMESTAMP — 16 H'FFFDAB2 16  MB[14]. CONTROL0_H — 16 H'FFFDAC2 16  LAFMO — 16 H'FFFDAC2 16  LAFM1 — 16 H'FFFDAC6 16  DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDACA 8, 16  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDAC 8, 16  TIMESTAMP — 16 H'FFFDAC2 16  MB[15]. CONTROL0_H — 16 H'FFFDAE2 16  CONTROL0_L — 16 H'FFFDAE2 16  CONTROL0_L — 16 H'FFFDAE2 16  LAFMO — 16 H'FFFFDAE2 16  |        |          | DATA_23    | _            | 16             | H'FFFFDAAA | 8, 16          |
| CONTROL1 — 16 H'FFFDAB0 8, 16  TIMESTAMP — 16 H'FFFDAB2 16  MB[14]. CONTROL0_H — 16 H'FFFDAC0 16, 32  CONTROL0_L — 16 H'FFFDAC2 16  LAFM0 — 16 H'FFFDAC4 16, 32  LAFM1 — 16 H'FFFDAC6 16  DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDACA 8, 16  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDAC8 8, 16  TIMESTAMP — 16 H'FFFDAD0 16, 32  CONTROL0_H — 16 H'FFFDAE2 16  MB[15]. CONTROL0_H — 16 H'FFFDAE2 16  CONTROL0_L — 16 H'FFFDAE2 16  LAFM0 — 16 H'FFFDAE2 16  |        |          | DATA_45    | _            | 16             | H'FFFFDAAC | 8, 16, 32      |
| TIMESTAMP — 16 H'FFFDAB2 16  MB[14]. CONTROLO_H — 16 H'FFFDAC0 16, 32  CONTROLO_L — 16 H'FFFDAC2 16  LAFM0 — 16 H'FFFDAC4 16, 32  LAFM1 — 16 H'FFFDAC6 16  DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDACA 8, 16  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDAD0 8, 16  TIMESTAMP — 16 H'FFFDAD2 16  MB[15]. CONTROLO_H — 16 H'FFFDAE2 16  LAFM0 — 16 H'FFFDAE2 16   |        |          | DATA_67    | _            | 16             | H'FFFFDAAE | 8, 16          |
| MB[14]. CONTROLO_H — 16 H'FFFDAC0 16, 32  CONTROLO_L — 16 H'FFFDAC2 16  LAFM0 — 16 H'FFFDAC4 16, 32  LAFM1 — 16 H'FFFDAC6 16  DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDACA 8, 16  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDACE 8, 16  TIMESTAMP — 16 H'FFFDAD2 16  MB[15]. CONTROLO_H — 16 H'FFFDAE2 16  LAFM0 — 16 H'FFFDAE2 16  |        |          | CONTROL1   | _            | 16             | H'FFFFDAB0 | 8, 16          |
| CONTROLO_L — 16 H'FFFDAC2 16  LAFM0 — 16 H'FFFDAC4 16, 32  LAFM1 — 16 H'FFFDAC6 16  DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDACA 8, 16  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDAD0 8, 16  TIMESTAMP — 16 H'FFFDAD2 16  MB[15]. CONTROLO_H — 16 H'FFFDAE2 16  LAFM0 — 16 H'FFFDAE4 16, 32  |        |          | TIMESTAMP  | _            | 16             | H'FFFFDAB2 | 16             |
| LAFM0 — 16 H'FFFDAC4 16, 32  LAFM1 — 16 H'FFFDAC6 16  DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDACA 8, 16  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDAD0 8, 16  TIMESTAMP — 16 H'FFFDAD2 16  MB[15]. CONTROL0_H — 16 H'FFFDAE0 16, 32  CONTROL0_L — 16 H'FFFDAE2 16  LAFM0 — 16 H'FFFDAE4 16, 32  |        | MB[14].  | CONTROL0_H | _            | 16             | H'FFFFDAC0 | 16, 32         |
| LAFM1 — 16 H'FFFDAC6 16  DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDACA 8, 16  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDAD0 8, 16  TIMESTAMP — 16 H'FFFDAD2 16  MB[15]. CONTROL0_H — 16 H'FFFDAE0 16, 32  CONTROL0_L — 16 H'FFFDAE2 16  LAFM0 — 16 H'FFFDAE4 16, 32   |        |          | CONTROL0_L | _            | 16             | H'FFFFDAC2 | 16             |
| DATA_01 — 16 H'FFFDAC8 8, 16, 32  DATA_23 — 16 H'FFFDACA 8, 16  DATA_45 — 16 H'FFFDACC 8, 16, 32  DATA_67 — 16 H'FFFDACE 8, 16  CONTROL1 — 16 H'FFFDAD0 8, 16  TIMESTAMP — 16 H'FFFDAD2 16  MB[15]. CONTROL0_H — 16 H'FFFDAE0 16, 32  CONTROL0_L — 16 H'FFFDAE2 16  LAFMO — 16 H'FFFDAE4 16, 32  |        |          | LAFM0      | _            | 16             | H'FFFFDAC4 | 16, 32         |
| DATA_23       —       16       H'FFFDACA 8, 16         DATA_45       —       16       H'FFFDACC 8, 16, 32         DATA_67       —       16       H'FFFDACE 8, 16         CONTROL1       —       16       H'FFFDADO 8, 16         TIMESTAMP       —       16       H'FFFDADO 16         MB[15].       CONTROLO_H       —       16       H'FFFDAEO 16, 32         CONTROLO_L       —       16       H'FFFDAE2 16         LAFMO       —       16       H'FFFFDAE4 16, 32  |        |          | LAFM1      | _            | 16             | H'FFFFDAC6 | 16             |
| DATA_45       —       16       H'FFFFDACC 8, 16, 32         DATA_67       —       16       H'FFFFDACE 8, 16         CONTROL1       —       16       H'FFFDAD0 8, 16         TIMESTAMP       —       16       H'FFFFDAD2 16         MB[15].       CONTROLO_H       —       16       H'FFFFDAE0 16, 32         CONTROLO_L       —       16       H'FFFFDAE4 16, 32   |        |          | DATA_01    | _            | 16             | H'FFFFDAC8 | 8, 16, 32      |
| DATA_67       —       16       H'FFFDACE 8, 16         CONTROL1       —       16       H'FFFDAD0 8, 16         TIMESTAMP       —       16       H'FFFDAD2 16         MB[15].       CONTROLO_H       —       16       H'FFFDAE0 16, 32         CONTROLO_L       —       16       H'FFFFDAE2 16         LAFM0       —       16       H'FFFFDAE4 16, 32   |        |          | DATA_23    | _            | 16             | H'FFFFDACA | 8, 16          |
| CONTROL1         —         16         H'FFFFDAD0         8, 16           TIMESTAMP         —         16         H'FFFFDAD2         16           MB[15].         CONTROLO_H         —         16         H'FFFFDAE0         16, 32           CONTROLO_L         —         16         H'FFFFDAE2         16           LAFMO         —         16         H'FFFFDAE4         16, 32   |        |          | DATA_45    | _            | 16             | H'FFFFDACC | 8, 16, 32      |
| TIMESTAMP       —       16       H'FFFDAD2       16         MB[15].       CONTROLO_H       —       16       H'FFFDAE0       16, 32         CONTROLO_L       —       16       H'FFFDAE2       16         LAFM0       —       16       H'FFFFDAE4       16, 32   |        |          | DATA_67    | _            | 16             | H'FFFFDACE | 8, 16          |
| MB[15]. CONTROLO_H — 16 H'FFFFDAE0 16, 32  CONTROLO_L — 16 H'FFFFDAE2 16  LAFMO — 16 H'FFFFDAE4 16, 32   |        |          | CONTROL1   | _            | 16             | H'FFFFDAD0 | 8, 16          |
| CONTROLO_L         —         16         H'FFFFDAE2         16           LAFM0         —         16         H'FFFFDAE4         16, 32   |        |          | TIMESTAMP  | _            | 16             | H'FFFFDAD2 | 16             |
| LAFM0 — 16 H'FFFFDAE4 16, 32   |        | MB[15].  | CONTROL0_H | _            | 16             | H'FFFFDAE0 | 16, 32         |
|  |        |          | CONTROL0_L | _            | 16             | H'FFFFDAE2 | 16             |
| LAFM1 — 16 H'FFFFDAE6 16   |        |          | LAFM0      | _            | 16             | H'FFFFDAE4 | 16, 32         |
|  |        |          | LAFM1      |              | 16             | H'FFFFDAE6 | 16             |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[15].  | DATA_01    | _            | 16             | H'FFFFDAE8 | 8, 16, 32      |
| TL1<br>(RCAN_ |          | DATA_23    | _            | 16             | H'FFFFDAEA | 8, 16          |
| B)            |          | DATA_45    | _            | 16             | H'FFFFDAEC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDAEE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDAF0 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFDAF2 | 16             |
|               | MB[16].  | CONTROL0_H | _            | 16             | H'FFFFDB00 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDB02 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDB04 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDB06 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDB08 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDB0A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDB0C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDB0E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDB10 | 8, 16          |
|               | MB[17].  | CONTROL0_H | _            | 16             | H'FFFFDB20 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDB22 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDB24 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDB26 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDB28 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDB2A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDB2C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDB2E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDB30 | 8, 16          |
|               | MB[18].  | CONTROL0_H | _            | 16             | H'FFFFDB40 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDB42 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDB44 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDB46 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDB48 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDB4A | 8, 16          |

| Module        | Register Name |            | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|---------------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[18].       | DATA_45    | _            | 16             | H'FFFFDB4C | 8, 16, 32      |
| TL1<br>(RCAN_ |               | DATA_67    | _            | 16             | H'FFFFDB4E | 8, 16          |
| B)            |               | CONTROL1   | _            | 16             | H'FFFFDB50 | 8, 16          |
|               | MB[19].       | CONTROL0_H | _            | 16             | H'FFFFDB60 | 16, 32         |
|               |               | CONTROL0_L | _            | 16             | H'FFFFDB62 | 16             |
|               |               | LAFM0      | _            | 16             | H'FFFFDB64 | 16, 32         |
|               |               | LAFM1      | _            | 16             | H'FFFFDB66 | 16             |
|               |               | DATA_01    | _            | 16             | H'FFFFDB68 | 8, 16, 32      |
|               |               | DATA_23    | _            | 16             | H'FFFFDB6A | 8, 16          |
|               |               | DATA_45    | _            | 16             | H'FFFFDB6C | 8, 16, 32      |
|               |               | DATA_67    | _            | 16             | H'FFFFDB6E | 8, 16          |
|               |               | CONTROL1   | _            | 16             | H'FFFFDB70 | 8, 16          |
|               | MB[20].       | CONTROL0_H | _            | 16             | H'FFFFDB80 | 16, 32         |
|               |               | CONTROL0_L | _            | 16             | H'FFFFDB82 | 16             |
|               |               | LAFM0      | _            | 16             | H'FFFFDB84 | 16, 32         |
|               |               | LAFM1      | _            | 16             | H'FFFFDB86 | 16             |
|               |               | DATA_01    | _            | 16             | H'FFFFDB88 | 8, 16, 32      |
|               |               | DATA_23    | _            | 16             | H'FFFFDB8A | 8, 16          |
|               |               | DATA_45    | _            | 16             | H'FFFFDB8C | 8, 16, 32      |
|               |               | DATA_67    | _            | 16             | H'FFFFDB8E | 8, 16          |
|               |               | CONTROL1   | _            | 16             | H'FFFFDB90 | 8, 16          |
|               | MB[21].       | CONTROL0_H | _            | 16             | H'FFFFDBA0 | 16, 32         |
|               |               | CONTROL0_L | _            | 16             | H'FFFFDBA2 | 16             |
|               |               | LAFM0      | _            | 16             | H'FFFFDBA4 | 16, 32         |
|               |               | LAFM1      | _            | 16             | H'FFFFDBA6 | 16             |
|               |               | DATA_01    | _            | 16             | H'FFFFDBA8 | 8, 16, 32      |
|               |               | DATA_23    | _            | 16             | H'FFFFDBAA | 8, 16          |
|               |               | DATA_45    | _            | 16             | H'FFFFDBAC | 8, 16, 32      |
|               |               | DATA_67    | _            | 16             | H'FFFFDBAE | 8, 16          |
|               |               | CONTROL1   |              | 16             | H'FFFFDBB0 | 8, 16          |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[22].  | CONTROL0_H | _            | 16             | H'FFFFDBC0 | 16, 32         |
| TL1<br>(RCAN_ |          | CONTROL0_L | _            | 16             | H'FFFFDBC2 | 16             |
| B)            |          | LAFM0      | _            | 16             | H'FFFFDBC4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDBC6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDBC8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDBCA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDBCC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDBCE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDBD0 | 8, 16          |
|               | MB[23].  | CONTROL0_H | _            | 16             | H'FFFFDBE0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDBE2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDBE4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDBE6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDBE8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDBEA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDBEC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDBEE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDBF0 | 8, 16          |
|               | MB[24].  | CONTROL0_H | _            | 16             | H'FFFFDC00 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDC02 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDC04 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDC06 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDC08 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDC0A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDC0C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDC0E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDC10 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFDC14 | 16             |
|               |          | TTCONTROL  |              | 16             | H'FFFFDC16 | 16             |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
|               | MB[25].  | CONTROL0_H | _            | 16             | H'FFFFDC20 | 16, 32         |
| TL1<br>(RCAN_ |          | CONTROL0_L | _            | 16             | H'FFFFDC22 | 16             |
| B)            |          | LAFM0      | _            | 16             | H'FFFFDC24 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDC26 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDC28 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDC2A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDC2C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDC2E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDC30 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFDC34 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFDC36 | 16             |
| -             | MB[26].  | CONTROL0_H | _            | 16             | H'FFFFDC40 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDC42 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDC44 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDC46 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDC48 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDC4A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDC4C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDC4E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDC50 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFDC54 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFDC56 | 16             |
| -             | MB[27].  | CONTROL0_H | _            | 16             | H'FFFFDC60 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDC62 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDC64 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDC66 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDC68 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDC6A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDC6C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDC6E | 8, 16          |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[27].  | CONTROL1   | _            | 16             | H'FFFFDC70 | 8, 16          |
| TL1<br>(RCAN_ |          | TTT        | _            | 16             | H'FFFFDC74 | 16             |
| B)            |          | TTCONTROL  | _            | 16             | H'FFFFDC76 | 16             |
|               | MB[28].  | CONTROL0_H | _            | 16             | H'FFFFDC80 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDC82 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDC84 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDC86 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDC88 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDC8A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDC8C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDC8E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDC90 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFDC94 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFDC96 | 16             |
|               | MB[29].  | CONTROL0_H | _            | 16             | H'FFFFDCA0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDCA2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDCA4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDCA6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFDCA8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFDCAA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFDCAC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFDCAE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFDCB0 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFDCB4 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFDCB6 | 16             |
|               | MB[30].  | CONTROL0_H | _            | 16             | H'FFFFDCC0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFDCC2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFDCC4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFDCC6 | 16             |
|               |          | DATA_01    |              | 16             | H'FFFFDCC8 | 8, 16, 32      |

| Module        | Register                              | Name                   | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|---------------------------------------|------------------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[30].                               | DATA_23                | _            | 16             | H'FFFFDCCA | 8, 16          |
| TL1<br>(RCAN_ |                                       | DATA_45                | _            | 16             | H'FFFFDCCC | 8, 16, 32      |
| B)            |                                       | DATA_67                | _            | 16             | H'FFFFDCCE | 8, 16          |
|               |                                       | CONTROL1               | _            | 16             | H'FFFFDCD0 | 8, 16          |
|               |                                       | TIMESTAMP              | _            | 16             | H'FFFFDCD2 | 16             |
|               |                                       | TTT                    | _            | 16             | H'FFFFDCD4 | 16             |
|               | MB[31].                               | CONTROL0_H             | _            | 16             | H'FFFFDCE0 | 16, 32         |
|               |                                       | CONTROL0_L             | _            | 16             | H'FFFFDCE2 | 16             |
|               |                                       | LAFM0                  | _            | 16             | H'FFFFDCE4 | 16, 32         |
|               |                                       | LAFM1                  | _            | 16             | H'FFFFDCE6 | 16             |
|               |                                       | DATA_01                | _            | 16             | H'FFFFDCE8 | 8, 16, 32      |
|               |                                       | DATA_23                | _            | 16             | H'FFFFDCEA | 8, 16          |
|               |                                       | DATA_45                | _            | 16             | H'FFFFDCEC | 8, 16, 32      |
|               |                                       | DATA_67                | _            | 16             | H'FFFFDCEE | 8, 16          |
|               |                                       | CONTROL1               | _            | 16             | H'FFFFDCF0 | 8, 16          |
|               |                                       | TIMESTAMP              | _            | 16             | H'FFFFDCF2 | 16             |
|               | Message buffer error status register  |                        | MBESR        | 16             | H'FFFFDE00 | 16             |
|               | Message buffer error control register |                        | MBECR        | 16             | H'FFFFDE02 | 16             |
| RCAN-         | Master control register               |                        | MCR          | 16             | H'FFFFE000 | 16             |
| TL1<br>(RCAN_ | General s                             | status register        | GSR          | 16             | H'FFFFE002 | 16             |
| (110A11_      | Bit config                            | uration register 1     | BCR1         | 16             | H'FFFFE004 | 16             |
|               | Bit config                            | uration register 0     | BCR0         | 16             | H'FFFFE006 | 16             |
|               | Interrupt                             | register               | IRR          | 16             | H'FFFFE008 | 16             |
|               | Interrupt                             | mask register          | IMR          | 16             | H'FFFFE00A | 16             |
|               | Error cou                             | nter register          | TEC/REC      | 16             | H'FFFFE00C | 16             |
|               | Transmit                              | pending register 1     | TXPR1        | 16             | H'FFFFE020 | 32             |
|               | Transmit                              | pending register 0     | TXPR0        | 16             | _          |                |
|               | Transmit                              | cancel register 1      | TXCR1        | 16             | H'FFFFE028 | 16             |
|               | Transmit                              | cancel register 0      | TXCR0        | 16             | H'FFFFE02A | 16             |
|               | Transmit                              | acknowledge register 1 | TXACK1       | 16             | H'FFFFE030 | 16             |

| Module        | Register                          | Name                       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|-----------------------------------|----------------------------|--------------|----------------|------------|----------------|
| RCAN-         | Transmit acknowledge register 0   |                            | TXACK0       | 16             | H'FFFFE032 | 16             |
| TL1<br>(RCAN_ | Abort ack                         | nowledge register 1        | ABACK1       | 16             | H'FFFFE038 | 16             |
| (110AN_       | Abort ack                         | nowledge register 0        | ABACK0       | 16             | H'FFFFE03A | 16             |
|               | Data fran<br>register 1           | ne receive pending         | RXPR1        | 16             | H'FFFFE040 | 16             |
|               | Data fran<br>register 0           | ne receive pending         | RXPR0        | 16             | H'FFFFE042 | 16             |
|               | Remote f<br>register 1            | rame receive pending       | RFPR1        | 16             | H'FFFFE048 | 16             |
|               | Remote f                          | rame receive pending       | RFPR0        | 16             | H'FFFFE04A | 16             |
|               | Mailbox interrupt mask register 1 |                            | MBIMR1       | 16             | H'FFFFE050 | 16             |
|               | Mailbox i                         | nterrupt mask register 0   | MBIMR0       | 16             | H'FFFFE052 | 16             |
|               | Unread message status register 1  |                            | UMSR1        | 16             | H'FFFFE058 | 16             |
|               | Unread n                          | nessage status             | UMSR0        | 16             | H'FFFFE05A | 16             |
|               | Timer trig                        | gger control register 0    | TTCR0        | 16             | H'FFFFE080 | 16             |
|               | Cycle ma                          | ximum/tx-enable<br>egister | CMAX_TEW     | 16             | H'FFFFE084 | 16             |
|               | Referenc                          | e trigger offset register  | RFTROFF      | 16             | H'FFFFE086 | 16             |
|               | Timer sta                         | tus register               | TSR          | 16             | H'FFFFE088 | 16             |
|               | Cycle cou                         | unter register             | CCR          | 16             | H'FFFFE08A | 16             |
|               | Timer co                          | unter register             | TCNTR        | 16             | H'FFFFE08C | 16             |
|               | Cycle tim                         | e register                 | CYCTR        | 16             | H'FFFFE090 | 16             |
|               | Referenc                          | e mark register            | RFMK         | 16             | H'FFFFE094 | 16             |
|               | Timer co                          | mpare match register 0     | TCMR0        | 16             | H'FFFFE098 | 16             |
|               | Timer co                          | mpare match register 1     | TCMR1        | 16             | H'FFFFE09C | 16             |
|               | Timer co                          | mpare match register 2     | TCMR2        | 16             | H'FFFFE0A0 | 16             |
|               | Tx-trigge                         | r time selection register  | TTTSEL       | 16             | H'FFFFE0A4 | 16             |
|               | MB[0].                            | CONTROL0_H                 |              | 16             | H'FFFFE100 | 16, 32         |
|               |                                   | CONTROL0_L                 | _            | 16             | H'FFFFE102 | 16             |
|               |                                   | LAFM0                      | _            | 16             | H'FFFFE104 | 16, 32         |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[0].   | LAFM1      | _            | 16             | H'FFFFE106 | 16             |
| TL1<br>(RCAN_ |          | DATA_01    | _            | 16             | H'FFFFE108 | 8, 16, 32      |
| (110AN_       |          | DATA_23    | _            | 16             | H'FFFFE10A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE10C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE10E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE110 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFE112 | 16             |
|               | MB[1].   | CONTROL0_H | _            | 16             | H'FFFFE120 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE122 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE124 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE126 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE128 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE12A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE12C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE12E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE130 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFE132 | 16             |
|               | MB[2].   | CONTROL0_H | _            | 16             | H'FFFFE140 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE142 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE144 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE146 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE148 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE14A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE14C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE14E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE150 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFE152 | 16             |
|               | MB[3].   | CONTROL0_H | _            | 16             | H'FFFFE160 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE162 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE164 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE166 | 16             |

| Module             | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--------------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-              | MB[3].   | DATA_01    | _            | 16             | H'FFFFE168 | 8, 16, 32      |
| TL1<br>(RCAN_      |          | DATA_23    | _            | 16             | H'FFFFE16A | 8, 16          |
| (110711 <b>1</b> _ |          | DATA_45    | _            | 16             | H'FFFFE16C | 8, 16, 32      |
|                    |          | DATA_67    | _            | 16             | H'FFFFE16E | 8, 16          |
|                    |          | CONTROL1   | _            | 16             | H'FFFFE170 | 8, 16          |
|                    |          | TIMESTAMP  | _            | 16             | H'FFFFE172 | 16             |
|                    | MB[4].   | CONTROL0_H | _            | 16             | H'FFFFE180 | 16, 32         |
|                    |          | CONTROL0_L | _            | 16             | H'FFFFE182 | 16             |
|                    |          | LAFM0      | _            | 16             | H'FFFFE184 | 16, 32         |
|                    |          | LAFM1      | _            | 16             | H'FFFFE186 | 16             |
|                    |          | DATA_01    | _            | 16             | H'FFFFE188 | 8, 16, 32      |
|                    |          | DATA_23    | _            | 16             | H'FFFFE18A | 8, 16          |
|                    |          | DATA_45    | _            | 16             | H'FFFFE18C | 8, 16, 32      |
|                    |          | DATA_67    | _            | 16             | H'FFFFE18E | 8, 16          |
|                    |          | CONTROL1   | _            | 16             | H'FFFFE190 | 8, 16          |
|                    |          | TIMESTAMP  | _            | 16             | H'FFFFE192 | 16             |
|                    | MB[5].   | CONTROL0_H | _            | 16             | H'FFFFE1A0 | 16, 32         |
|                    |          | CONTROL0_L | _            | 16             | H'FFFFE1A2 | 16             |
|                    |          | LAFM0      | _            | 16             | H'FFFFE1A4 | 16, 32         |
|                    |          | LAFM1      | _            | 16             | H'FFFFE1A6 | 16             |
|                    |          | DATA_01    | _            | 16             | H'FFFFE1A8 | 8, 16, 32      |
|                    |          | DATA_23    | _            | 16             | H'FFFFE1AA | 8, 16          |
|                    |          | DATA_45    | _            | 16             | H'FFFFE1AC | 8, 16, 32      |
|                    |          | DATA_67    | _            | 16             | H'FFFFE1AE | 8, 16          |
|                    |          | CONTROL1   | _            | 16             | H'FFFFE1B0 | 8, 16          |
|                    |          | TIMESTAMP  | _            | 16             | H'FFFFE1B2 | 16             |
|                    | MB[6].   | CONTROL0_H | _            | 16             | H'FFFFE1C0 | 16, 32         |
|                    |          | CONTROL0_L | _            | 16             | H'FFFFE1C2 | 16             |
|                    |          | LAFM0      | _            | 16             | H'FFFFE1C4 | 16, 32         |
|                    |          | LAFM1      | _            | 16             | H'FFFFE1C6 | 16             |

| RCAN-TL1 (RCAN_C)  | Module | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|--|--------|----------|------------|--------------|----------------|------------|----------------|
| RCAN_C    DATA_45  |        | MB[6].   | DATA_01    | _            | 16             | H'FFFFE1C8 | 8, 16, 32      |
| DATA_45 — 16 H'FFFFE1CC 8, 16, 32  DATA_67 — 16 H'FFFFE1CE 8, 16  CONTROL1 — 16 H'FFFFE1D0 8, 16  TIMESTAMP — 16 H'FFFFE1D1 16  MB[7]. CONTROL0_H — 16 H'FFFFE1E0 16, 32  CONTROL0_L — 16 H'FFFFE1E0 16, 32  LAFM0 — 16 H'FFFFE1E0 16  DATA_01 — 16 H'FFFFE1E0 16  DATA_23 — 16 H'FFFFE1E0 8, 16  DATA_45 — 16 H'FFFFE1E0 8, 16  DATA_67 — 16 H'FFFFE1E0 8, 16  CONTROL1 — 16 H'FFFFE1E0 8, 16  TIMESTAMP — 16 H'FFFFE1E1 8, 16  MB[8]. CONTROL0_H — 16 H'FFFFE1E2 16  MB[8]. CONTROL0_H — 16 H'FFFFE200 16, 32  CONTROL0_L — 16 H'FFFFE200 16, 32  LAFM0 — 16 H'FFFFE200 16, 32  LAFM0 — 16 H'FFFFE200 16   |        |          | DATA_23    | _            | 16             | H'FFFFE1CA | 8, 16          |
| CONTROL1 — 16 H'FFFE1D0 8, 16  TIMESTAMP — 16 H'FFFFE1D2 16  MB[7]. CONTROL0_H — 16 H'FFFFE1E0 16, 32  CONTROL0_L — 16 H'FFFFE1E2 16  LAFM0 — 16 H'FFFFE1E4 16, 32  LAFM1 — 16 H'FFFFE1E6 16  DATA_01 — 16 H'FFFFE1E8 8, 16, 32  DATA_23 — 16 H'FFFFE1EA 8, 16  DATA_45 — 16 H'FFFFE1EC 8, 16, 32  DATA_67 — 16 H'FFFFE1E 8, 16  CONTROL1 — 16 H'FFFFE1E 8, 16  TIMESTAMP — 16 H'FFFFE1F0 16, 32  MB[8]. CONTROL0_H — 16 H'FFFFE200 16, 32  CONTROL0_L — 16 H'FFFFE202 16  LAFM0 — 16 H'FFFFE204 16, 32  LAFM1 — 16 H'FFFFE206 16  | •      |          | DATA_45    | _            | 16             | H'FFFFE1CC | 8, 16, 32      |
| TIMESTAMP — 16 H'FFFE1D2 16  MB[7]. CONTROLO_H — 16 H'FFFFE1E0 16, 32  CONTROLO_L — 16 H'FFFFE1E2 16  LAFM0 — 16 H'FFFFE1E4 16, 32  LAFM1 — 16 H'FFFFE1E6 16  DATA_01 — 16 H'FFFFE1E8 8, 16, 32  DATA_23 — 16 H'FFFFE1EA 8, 16  DATA_45 — 16 H'FFFFE1EC 8, 16, 32  DATA_67 — 16 H'FFFFE1E 8, 16  CONTROL1 — 16 H'FFFFE1E 8, 16  TIMESTAMP — 16 H'FFFFE1F0 8, 16  MB[8]. CONTROLO_H — 16 H'FFFFE200 16, 32  CONTROLO_L — 16 H'FFFFE204 16, 32  LAFM0 — 16 H'FFFFE206 16   |        |          | DATA_67    | _            | 16             | H'FFFFE1CE | 8, 16          |
| MB[7].   |        |          | CONTROL1   | _            | 16             | H'FFFFE1D0 | 8, 16          |
| CONTROLO_L — 16 H'FFFFE1E2 16  LAFM0 — 16 H'FFFFE1E4 16, 32  LAFM1 — 16 H'FFFFE1E6 16  DATA_01 — 16 H'FFFFE1E8 8, 16, 32  DATA_23 — 16 H'FFFFE1EA 8, 16  DATA_45 — 16 H'FFFFE1EC 8, 16, 32  DATA_67 — 16 H'FFFFE1E 8, 16  CONTROL1 — 16 H'FFFFE1F0 8, 16  TIMESTAMP — 16 H'FFFFE1F2 16  MB[8]. CONTROLO_H — 16 H'FFFFE200 16, 32  CONTROLO_L — 16 H'FFFFE202 16  LAFM0 — 16 H'FFFFE204 16, 32  LAFM1 — 16 H'FFFFE206 16  |        |          | TIMESTAMP  | _            | 16             | H'FFFFE1D2 | 16             |
| LAFM0 — 16 H'FFFE1E4 16, 32  LAFM1 — 16 H'FFFE1E6 16  DATA_01 — 16 H'FFFE1E8 8, 16, 32  DATA_23 — 16 H'FFFFE1EA 8, 16  DATA_45 — 16 H'FFFFE1EC 8, 16, 32  DATA_67 — 16 H'FFFFE1E 8, 16  CONTROL1 — 16 H'FFFFE1F0 8, 16  TIMESTAMP — 16 H'FFFFE1F2 16  MB[8]. CONTROL0_H — 16 H'FFFFE200 16, 32  CONTROL0_L — 16 H'FFFFE204 16, 32  LAFM0 — 16 H'FFFFE206 16  |        | MB[7].   | CONTROL0_H | _            | 16             | H'FFFFE1E0 | 16, 32         |
| LAFM1 — 16 H'FFFFE1E6 16  DATA_01 — 16 H'FFFFE1E8 8, 16, 32  DATA_23 — 16 H'FFFFE1EA 8, 16  DATA_45 — 16 H'FFFFE1EC 8, 16, 32  DATA_67 — 16 H'FFFFE1EE 8, 16  CONTROL1 — 16 H'FFFFE1F0 8, 16  TIMESTAMP — 16 H'FFFFE1F2 16  MB[8]. CONTROL0_H — 16 H'FFFFE200 16, 32  CONTROL0_L — 16 H'FFFFE202 16  LAFM0 — 16 H'FFFFE204 16, 32  LAFM1 — 16 H'FFFFE206 16  |        |          | CONTROL0_L | _            | 16             | H'FFFFE1E2 | 16             |
| DATA_01       —       16       H'FFFFE1E8       8, 16, 32         DATA_23       —       16       H'FFFFE1EA       8, 16         DATA_45       —       16       H'FFFFE1EC       8, 16, 32         DATA_67       —       16       H'FFFFE1EE       8, 16         CONTROL1       —       16       H'FFFFE1F0       8, 16         TIMESTAMP       —       16       H'FFFFE1F2       16         MB[8].       CONTROLO_H       —       16       H'FFFFE200       16, 32         CONTROLO_L       —       16       H'FFFFE204       16, 32         LAFM0       —       16       H'FFFFE206       16         LAFM1       —       16       H'FFFFE206       16 |        |          | LAFM0      | _            | 16             | H'FFFFE1E4 | 16, 32         |
| DATA_23 — 16 H'FFFFE1EA 8, 16  DATA_45 — 16 H'FFFFE1EC 8, 16, 32  DATA_67 — 16 H'FFFFE1EE 8, 16  CONTROL1 — 16 H'FFFFE1F0 8, 16  TIMESTAMP — 16 H'FFFFE1F2 16  MB[8]. CONTROL0_H — 16 H'FFFFE200 16, 32  CONTROL0_L — 16 H'FFFFE202 16  LAFM0 — 16 H'FFFFE204 16, 32  LAFM1 — 16 H'FFFFE206 16   |        |          | LAFM1      | _            | 16             | H'FFFFE1E6 | 16             |
| DATA_45 — 16 H'FFFFE1EC 8, 16, 32  DATA_67 — 16 H'FFFFE1EE 8, 16  CONTROL1 — 16 H'FFFFE1F0 8, 16  TIMESTAMP — 16 H'FFFFE1F2 16  MB[8]. CONTROL0_H — 16 H'FFFFE200 16, 32  CONTROL0_L — 16 H'FFFFE202 16  LAFM0 — 16 H'FFFFE204 16, 32  LAFM1 — 16 H'FFFFE206 16  |        |          | DATA_01    | _            | 16             | H'FFFFE1E8 | 8, 16, 32      |
| DATA_67       —       16       H'FFFFE1EE       8, 16         CONTROL1       —       16       H'FFFFE1F0       8, 16         TIMESTAMP       —       16       H'FFFFE1F2       16         MB[8].       CONTROLO_H       —       16       H'FFFFE200       16, 32         CONTROLO_L       —       16       H'FFFFE202       16         LAFM0       —       16       H'FFFFE204       16, 32         LAFM1       —       16       H'FFFFE206       16   |        |          | DATA_23    | _            | 16             | H'FFFFE1EA | 8, 16          |
| CONTROL1 — 16 H'FFFFE1F0 8, 16  TIMESTAMP — 16 H'FFFFE1F2 16  MB[8]. CONTROL0_H — 16 H'FFFFE200 16, 32  CONTROL0_L — 16 H'FFFFE202 16  LAFM0 — 16 H'FFFFE204 16, 32  LAFM1 — 16 H'FFFFE206 16  |        |          | DATA_45    | _            | 16             | H'FFFFE1EC | 8, 16, 32      |
| TIMESTAMP         —         16         H'FFFFE1F2         16           MB[8].         CONTROLO_H         —         16         H'FFFFE200         16, 32           CONTROLO_L         —         16         H'FFFFE202         16           LAFM0         —         16         H'FFFFE204         16, 32           LAFM1         —         16         H'FFFFE206         16  |        |          | DATA_67    | _            | 16             | H'FFFFE1EE | 8, 16          |
| MB[8]. CONTROLO_H — 16 H'FFFFE200 16, 32  CONTROLO_L — 16 H'FFFFE202 16  LAFM0 — 16 H'FFFFE204 16, 32  LAFM1 — 16 H'FFFFE206 16  |        |          | CONTROL1   | _            | 16             | H'FFFFE1F0 | 8, 16          |
| CONTROLO_L         —         16         H'FFFFE202         16           LAFM0         —         16         H'FFFFE204         16, 32           LAFM1         —         16         H'FFFFE206         16  |        |          | TIMESTAMP  | _            | 16             | H'FFFFE1F2 | 16             |
| LAFM0 — 16 H'FFFFE204 16, 32  LAFM1 — 16 H'FFFFE206 16   |        | MB[8].   | CONTROL0_H | _            | 16             | H'FFFFE200 | 16, 32         |
| LAFM1 — 16 H'FFFFE206 16   |        |          | CONTROL0_L | _            | 16             | H'FFFFE202 | 16             |
|  |        |          | LAFM0      | _            | 16             | H'FFFFE204 | 16, 32         |
| DATA_01 — 16 H'FFFFE208 8, 16, 32  |        |          | LAFM1      | _            | 16             | H'FFFFE206 | 16             |
|  |        |          | DATA_01    | _            | 16             | H'FFFFE208 | 8, 16, 32      |
| DATA_23 — 16 H'FFFFE20A 8, 16  |        |          | DATA_23    | _            | 16             | H'FFFFE20A | 8, 16          |
| DATA_45 — 16 H'FFFFE20C 8, 16, 32  |        |          | DATA_45    | _            | 16             | H'FFFFE20C | 8, 16, 32      |
| DATA_67 — 16 H'FFFFE20E 8, 16  |        |          | DATA_67    | _            | 16             | H'FFFFE20E | 8, 16          |
| CONTROL1 — 16 H'FFFFE210 8, 16   |        |          | CONTROL1   | _            | 16             | H'FFFFE210 | 8, 16          |
| TIMESTAMP — 16 H'FFFFE212 16   |        |          | TIMESTAMP  | _            | 16             | H'FFFFE212 | 16             |
| MB[9]. CONTROL0_H — 16 H'FFFFE220 16, 32   |        | MB[9].   | CONTROL0_H | _            | 16             | H'FFFFE220 | 16, 32         |
| CONTROLO_L — 16 H'FFFFE222 16  |        |          | CONTROL0_L | _            | 16             | H'FFFFE222 | 16             |
| LAFM0 — 16 H'FFFFE224 16, 32   |        |          | LAFM0      | _            | 16             | H'FFFFE224 | 16, 32         |
| LAFM1 — 16 H'FFFFE226 16   |        |          | LAFM1      |              | 16             | H'FFFFE226 | 16             |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[9].   | DATA_01    | _            | 16             | H'FFFFE228 | 8, 16, 32      |
| TL1<br>(RCAN_ |          | DATA_23    | _            | 16             | H'FFFFE22A | 8, 16          |
| C)            |          | DATA_45    | _            | 16             | H'FFFFE22C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE22E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE230 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFE232 | 16             |
|               | MB[10].  | CONTROL0_H | _            | 16             | H'FFFFE240 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE242 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE244 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE246 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE248 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE24A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE24C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE24E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE250 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFE252 | 16             |
|               | MB[11].  | CONTROL0_H | _            | 16             | H'FFFFE260 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE262 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE264 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE266 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE268 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE26A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE26C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE26E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE270 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFE272 | 16             |
|               | MB[12].  | CONTROL0_H |              | 16             | H'FFFFE280 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE282 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE284 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE286 | 16             |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[12].  | DATA_01    | _            | 16             | H'FFFFE288 | 8, 16, 32      |
| TL1<br>(RCAN_ |          | DATA_23    | _            | 16             | H'FFFFE28A | 8, 16          |
| (110AN_<br>C) |          | DATA_45    | _            | 16             | H'FFFFE28C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE28E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE290 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFE292 | 16             |
| •             | MB[13].  | CONTROL0_H | _            | 16             | H'FFFFE2A0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE2A2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE2A4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE2A6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE2A8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE2AA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE2AC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE2AE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE2B0 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFE2B2 | 16             |
| •             | MB[14].  | CONTROL0_H | _            | 16             | H'FFFFE2C0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE2C2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE2C4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE2C6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE2C8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE2CA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE2CC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE2CE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE2D0 | 8, 16          |
|               |          | TIMESTAMP  | _            | 16             | H'FFFFE2D2 | 16             |
|               | MB[15].  | CONTROL0_H | _            | 16             | H'FFFFE2E0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE2E2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE2E4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE2E6 | 16             |

| Module            | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|-------------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-             | MB[15].  | DATA_01    | _            | 16             | H'FFFFE2E8 | 8, 16, 32      |
| TL1<br>(RCAN_     |          | DATA_23    | <del>_</del> | 16             | H'FFFFE2EA | 8, 16          |
| (1107111 <u> </u> |          | DATA_45    | _            | 16             | H'FFFFE2EC | 8, 16, 32      |
|                   |          | DATA_67    | _            | 16             | H'FFFFE2EE | 8, 16          |
|                   |          | CONTROL1   | _            | 16             | H'FFFFE2F0 | 8, 16          |
|                   |          | TIMESTAMP  | _            | 16             | H'FFFFE2F2 | 16             |
|                   | MB[16].  | CONTROL0_H | _            | 16             | H'FFFFE300 | 16, 32         |
|                   |          | CONTROL0_L | _            | 16             | H'FFFFE302 | 16             |
|                   |          | LAFM0      | _            | 16             | H'FFFFE304 | 16, 32         |
|                   |          | LAFM1      | _            | 16             | H'FFFFE306 | 16             |
|                   |          | DATA_01    | <del>_</del> | 16             | H'FFFFE308 | 8, 16, 32      |
|                   |          | DATA_23    | _            | 16             | H'FFFFE30A | 8, 16          |
|                   |          | DATA_45    | _            | 16             | H'FFFFE30C | 8, 16, 32      |
|                   |          | DATA_67    | <del>_</del> | 16             | H'FFFFE30E | 8, 16          |
|                   |          | CONTROL1   | _            | 16             | H'FFFFE310 | 8, 16          |
|                   | MB[17].  | CONTROL0_H | _            | 16             | H'FFFFE320 | 16, 32         |
|                   |          | CONTROL0_L | _            | 16             | H'FFFFE322 | 16             |
|                   |          | LAFM0      | _            | 16             | H'FFFFE324 | 16, 32         |
|                   |          | LAFM1      | _            | 16             | H'FFFFE326 | 16             |
|                   |          | DATA_01    | _            | 16             | H'FFFFE328 | 8, 16, 32      |
|                   |          | DATA_23    | _            | 16             | H'FFFFE32A | 8, 16          |
|                   |          | DATA_45    | _            | 16             | H'FFFFE32C | 8, 16, 32      |
|                   |          | DATA_67    | _            | 16             | H'FFFFE32E | 8, 16          |
|                   |          | CONTROL1   | _            | 16             | H'FFFFE330 | 8, 16          |
|                   | MB[18].  | CONTROL0_H | _            | 16             | H'FFFFE340 | 16, 32         |
|                   |          | CONTROL0_L | _            | 16             | H'FFFFE342 | 16             |
|                   |          | LAFM0      | _            | 16             | H'FFFFE344 | 16, 32         |
|                   |          | LAFM1      | _            | 16             | H'FFFFE346 | 16             |
|                   |          | DATA_01    | _            | 16             | H'FFFFE348 | 8, 16, 32      |
|                   |          | DATA_23    | _            | 16             | H'FFFFE34A | 8, 16          |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[18].  | DATA_45    | _            | 16             | H'FFFFE34C | 8, 16, 32      |
| TL1<br>(RCAN_ |          | DATA_67    | _            | 16             | H'FFFFE34E | 8, 16          |
| C)            |          | CONTROL1   | _            | 16             | H'FFFFE350 | 8, 16          |
|               | MB[19].  | CONTROL0_H | _            | 16             | H'FFFFE360 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE362 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE364 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE366 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE368 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE36A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE36C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE36E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE370 | 8, 16          |
|               | MB[20].  | CONTROL0_H | _            | 16             | H'FFFFE380 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE382 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE384 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE386 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE388 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE38A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE38C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE38E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE390 | 8, 16          |
|               | MB[21].  | CONTROL0_H | _            | 16             | H'FFFFE3A0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE3A2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE3A4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE3A6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE3A8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE3AA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE3AC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE3AE | 8, 16          |
|               |          | CONTROL1   |              | 16             | H'FFFFE3B0 | 8, 16          |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[22].  | CONTROL0_H | _            | 16             | H'FFFFE3C0 | 16, 32         |
| TL1<br>(RCAN_ |          | CONTROL0_L | _            | 16             | H'FFFFE3C2 | 16             |
| (110AN_       |          | LAFM0      | _            | 16             | H'FFFFE3C4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE3C6 | 16             |
|               |          | DATA_01    |              | 16             | H'FFFFE3C8 | 8, 16, 32      |
|               |          | DATA_23    |              | 16             | H'FFFFE3CA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE3CC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE3CE | 8, 16          |
|               |          | CONTROL1   |              | 16             | H'FFFFE3D0 | 8, 16          |
|               | MB[23].  | CONTROL0_H | _            | 16             | H'FFFFE3E0 | 16, 32         |
|               |          | CONTROL0_L |              | 16             | H'FFFFE3E2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE3E4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE3E6 | 16             |
|               |          | DATA_01    |              | 16             | H'FFFFE3E8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE3EA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE3EC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE3EE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE3F0 | 8, 16          |
|               | MB[24].  | CONTROL0_H | _            | 16             | H'FFFFE400 | 16, 32         |
|               |          | CONTROL0_L |              | 16             | H'FFFFE402 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE404 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE406 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE408 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE40A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE40C | 8, 16, 32      |
|               |          | DATA_67    |              | 16             | H'FFFFE40E | 8, 16          |
|               |          | CONTROL1   |              | 16             | H'FFFFE410 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFE414 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFE416 | 16             |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[25].  | CONTROL0_H | _            | 16             | H'FFFFE420 | 16, 32         |
| TL1<br>(RCAN_ |          | CONTROL0_L | _            | 16             | H'FFFFE422 | 16             |
| (110A11_      |          | LAFM0      | _            | 16             | H'FFFFE424 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE426 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE428 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE42A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE42C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE42E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE430 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFE434 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFE436 | 16             |
|               | MB[26].  | CONTROL0_H | _            | 16             | H'FFFFE440 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE442 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE444 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE446 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE448 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE44A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE44C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE44E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE450 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFE454 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFE456 | 16             |
|               | MB[27].  | CONTROL0_H | _            | 16             | H'FFFFE460 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE462 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE464 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE466 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE468 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE46A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE46C | 8, 16, 32      |
|               |          | DATA_67    |              | 16             | H'FFFFE46E | 8, 16          |

| Module        | Register | Name       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|----------|------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[27].  | CONTROL1   | _            | 16             | H'FFFFE470 | 8, 16          |
| TL1<br>(RCAN_ |          | TTT        | _            | 16             | H'FFFFE474 | 16             |
| (110AN_       |          | TTCONTROL  | _            | 16             | H'FFFFE476 | 16             |
|               | MB[28].  | CONTROL0_H | _            | 16             | H'FFFFE480 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE482 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE484 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE486 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE488 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE48A | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE48C | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE48E | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE490 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFE494 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFE496 | 16             |
|               | MB[29].  | CONTROL0_H | _            | 16             | H'FFFFE4A0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE4A2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE4A4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE4A6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE4A8 | 8, 16, 32      |
|               |          | DATA_23    | _            | 16             | H'FFFFE4AA | 8, 16          |
|               |          | DATA_45    | _            | 16             | H'FFFFE4AC | 8, 16, 32      |
|               |          | DATA_67    | _            | 16             | H'FFFFE4AE | 8, 16          |
|               |          | CONTROL1   | _            | 16             | H'FFFFE4B0 | 8, 16          |
|               |          | TTT        | _            | 16             | H'FFFFE4B4 | 16             |
|               |          | TTCONTROL  | _            | 16             | H'FFFFE4B6 | 16             |
|               | MB[30].  | CONTROL0_H | _            | 16             | H'FFFFE4C0 | 16, 32         |
|               |          | CONTROL0_L | _            | 16             | H'FFFFE4C2 | 16             |
|               |          | LAFM0      | _            | 16             | H'FFFFE4C4 | 16, 32         |
|               |          | LAFM1      | _            | 16             | H'FFFFE4C6 | 16             |
|               |          | DATA_01    | _            | 16             | H'FFFFE4C8 | 8, 16, 32      |

| Module        | Register         | Name                | Abbreviation | Number of Bits | Address    | Access<br>Size |
|---------------|------------------|---------------------|--------------|----------------|------------|----------------|
| RCAN-         | MB[30].          | DATA_23             | _            | 16             | H'FFFFE4CA | 8, 16          |
| TL1<br>(RCAN_ |                  | DATA_45             | _            | 16             | H'FFFFE4CC | 8, 16, 32      |
| (110A11_      |                  | DATA_67             | _            | 16             | H'FFFFE4CE | 8, 16          |
|               |                  | CONTROL1            | _            | 16             | H'FFFFE4D0 | 8, 16          |
|               |                  | TIMESTAMP           | _            | 16             | H'FFFFE4D2 | 16             |
|               |                  | TTT                 | _            | 16             | H'FFFFE4D4 | 16             |
|               | MB[31].          | CONTROL0_H          | _            | 16             | H'FFFFE4E0 | 16, 32         |
|               |                  | CONTROL0_L          | _            | 16             | H'FFFFE4E2 | 16             |
|               |                  | LAFM0               | _            | 16             | H'FFFFE4E4 | 16, 32         |
|               |                  | LAFM1               | _            | 16             | H'FFFFE4E6 | 16             |
|               |                  | DATA_01             | _            | 16             | H'FFFFE4E8 | 8, 16, 32      |
|               |                  | DATA_23             | _            | 16             | H'FFFFE4EA | 8, 16          |
|               |                  | DATA_45             | _            | 16             | H'FFFFE4EC | 8, 16, 32      |
|               |                  | DATA_67             | _            | 16             | H'FFFFE4EE | 8, 16          |
|               |                  | CONTROL1            | _            | 16             | H'FFFFE4F0 | 8, 16          |
|               |                  | TIMESTAMP           | _            | 16             | H'FFFFE4F2 | 16             |
|               | Message register | buffer error status | MBESR        | 16             | H'FFFFE600 | 16             |
|               | Message register | buffer error status | MBECR        | 16             | H'FFFFE602 | 16             |
| ADC           | A/D data         | register Diag0      | ADRD0        | 16             | H'FFFFE83E | 16             |
| (ADC_A)       | A/D data         | register 0          | ADR0         | 16             | H'FFFFE840 | 16             |
|               | A/D data         | register 1          | ADR1         | 16             | H'FFFFE842 | 16             |
|               | A/D data         | register 2          | ADR2         | 16             | H'FFFFE844 | 16             |
|               | A/D data         | register 3          | ADR3         | 16             | H'FFFFE846 | 16             |
|               | A/D data         | register 4          | ADR4         | 16             | H'FFFFE848 | 16             |
|               | A/D data         | register 5          | ADR5         | 16             | H'FFFFE84A | 16             |
|               | A/D data         | register 6          | ADR6         | 16             | H'FFFFE84C | 16             |
|               | A/D data         | register 7          | ADR7         | 16             | H'FFFFE84E | 16             |
|               | A/D data         | register 8          | ADR8         | 16             | H'FFFFE850 | 16             |
|               | A/D data         | register 9          | ADR9         | 16             | H'FFFFE852 | 16             |
|               | A/D data         | register 10         | ADR10        | 16             | H'FFFFE854 | 16             |

| Module         | Register Name           | Abbreviation | Number of Bits | Address    | Access<br>Size |
|----------------|-------------------------|--------------|----------------|------------|----------------|
| ADC            | A/D data register 11    | ADR11        | 16             | H'FFFFE856 | 16             |
| (ADC_A)        | A/D data register 12    | ADR12        | 16             | H'FFFFE858 | 16             |
|                | A/D data register 13    | ADR13        | 16             | H'FFFFE85A | 16             |
|                | A/D data register 14    | ADR14        | 16             | H'FFFFE85C | 16             |
|                | A/D data register 15    | ADR15        | 16             | H'FFFFE85E | 16             |
|                | A/D data register 16    | ADR16        | 16             | H'FFFFE860 | 16             |
|                | A/D data register 17    | ADR17        | 16             | H'FFFFE862 | 16             |
|                | A/D data register 18    | ADR18        | 16             | H'FFFFE864 | 16             |
|                | A/D data register 19    | ADR19        | 16             | H'FFFFE866 | 16             |
|                | A/D data register 20    | ADR20        | 16             | H'FFFFE868 | 16             |
|                | A/D data register 21    | ADR21        | 16             | H'FFFFE86A | 16             |
|                | A/D data register 22    | ADR22        | 16             | H'FFFFE86C | 16             |
|                | A/D data register 23    | ADR23        | 16             | H'FFFFE86E | 16             |
|                | A/D data register 24    | ADR24        | 16             | H'FFFFE870 | 16             |
|                | A/D data register 25    | ADR25        | 16             | H'FFFFE872 | 16             |
|                | A/D data register 26    | ADR26        | 16             | H'FFFFE874 | 16             |
|                | A/D data register 27    | ADR27        | 16             | H'FFFFE876 | 16             |
| ADC            | A/D data register Diag1 | ADRD1        | 16             | H'FFFFEC3E | 16             |
| (ADC_B)        | A/D data register 40    | ADR40        | 16             | H'FFFFEC40 | 16             |
|                | A/D data register 41    | ADR41        | 16             | H'FFFFEC42 | 16             |
|                | A/D data register 42    | ADR42        | 16             | H'FFFFEC44 | 16             |
|                | A/D data register 43    | ADR43        | 16             | H'FFFFEC46 | 16             |
|                | A/D data register 44    | ADR44        | 16             | H'FFFFEC48 | 16             |
|                | A/D data register 45    | ADR45        | 16             | H'FFFFEC4A | 16             |
|                | A/D data register 46    | ADR46        | 16             | H'FFFFEC4C | 16             |
|                | A/D data register 47    | ADR47        | 16             | H'FFFFEC4E | 16             |
|                | A/D data register 48    | ADR48        | 16             | H'FFFFEC50 | 16             |
| ADC<br>(ADC_A) | A/D control register 0  | ADCSR0       | 8              | H'FFFFE800 | 8              |
| ADC<br>(ADC_B) | A/D control register 1  | ADCSR1       | 8              | H'FFFFEC00 | 8              |

| Module         | Register Name  | Abbreviation | Number of Bits | Address    | Access<br>Size |
|----------------|--|--------------|----------------|------------|----------------|
| ADC<br>(ADC_A) | A/D conversion status register 0                                 | ADREF0       | 8              | H'FFFFE802 | 8              |
| ADC<br>(ADC_B) | A/D conversion status register 1                                 | ADREF1       | 8              | H'FFFFEC02 | 8              |
| ADC<br>(ADC_A) | A/D interrupt trigger enable register 0                          | ADTRE0       | 16             | H'FFFFE804 | 8, 16          |
| ADC<br>(ADC_B) | A/D interrupt trigger enable register 1                          | ADTRE1       | 8              | H'FFFFEC10 | 8              |
| ADC<br>(ADC_A) | A/D interrupt trigger conversion end flag register0              | ADTRF0       | 16             | H'FFFFE806 | 8, 16          |
| ADC<br>(ADC_B) | A/D interrupt trigger conversion end flag register1              | ADTRF1       | 8              | H'FFFFEC12 | 8              |
| ADC<br>(ADC_A) |  |              | H'FFFFE808     | 8, 16      |                |
| ADC<br>(ADC_B) | A/D interrupt trigger source select register 1                   | ADTRS1       | 8              | H'FFFFEC14 | 8              |
| ADC<br>(ADC_A) | A/D interrupt software trigger register 0                        | ADSTRG0      | 16             | H'FFFFE80A | 8, 16          |
| ADC<br>(ADC_B) | A/D interrupt software trigger register 1                        | ADSTRG1      | 8              | H'FFFFEC16 | 8              |
| ADC<br>(ADC_A) | A/D interrupt trigger conversion end interrupt enable register 0 | ADTRD0       | 16             | H'FFFFE80C | 8, 16          |
| ADC<br>(ADC_B) | A/D interrupt trigger conversion end interrupt enable register 1 | ADTRD1       | 8              | H'FFFFEC18 | 8              |
| ADC<br>(ADC_A) | A/D-converted value addition mode select register 0              | ADADS0       | 8              | H'FFFFE81C | 8              |
| ADC<br>(ADC_B) | A/D-converted value addition mode select register 1              | ADADS1       | 8              | H'FFFFEC1C | 8              |
| ADC<br>(ADC_A) | A/D-converted value addition count selection register 0          | ADADC0       | 8              | H'FFFFE81E | 8              |
| ADC<br>(ADC_B) | A/D-converted value addition count selection register 1          | ADADC1       | 8              | H'FFFFEC1E | 8              |
| ADC            | A/D channel select register 0                                    | ADANS0       | 16             | H'FFFFE820 | 8, 16, 32      |
| (ADC_A)        | A/D channel select register 1                                    | ADANS1       | 16             | H'FFFFE822 | 8, 16, 32      |
| ADC<br>(ADC_B) | A/D channel select register 3                                    | ADANS3       | 16             | H'FFFFEC20 | 8, 16          |

| Module         | Register Name                       | Abbreviation | Number of Bits | Address    | Access<br>Size |
|----------------|-------------------------------------|--------------|----------------|------------|----------------|
| ADC<br>(ADC_A) | A/D control extended register 0     | ADCER0       | 16             | H'FFFFE830 | 8, 16          |
| ADC<br>(ADC_B) | A/D control extended register 1     | ADCER1       | 16             | H'FFFFEC30 | 8, 16          |
| JTAG           | Instruction register                | SDIR         | 4              | _          | _              |
|                | ID register                         | SDID         | 32             | _          | _              |
|                | Bypass register                     | SDBPR        | 1              | _          | _              |
|                | Boundary scan register              | SDBSR        | _              | _          | _              |
| AUD-II         | AUD control register                | AUCSR        | 16             | H'FFFC0C00 | 8, 16          |
|                | AUD window A start address register | AUWASR       | 32             | H'FFFC0C04 | 8, 16, 32      |
|                | AUD window A end address register   | AUWAER       | 32             | H'FFFC0C08 | 8, 16, 32      |
|                | AUD window B start address register | AUWBSR       | 32             | H'FFFC0C0C | 8, 16, 32      |
|                | AUD window B end address register   | AUWBER       | 32             | H'FFFC0C10 | 8, 16, 32      |
|                | AUD extended control register       | AUECSR       | 16             | H'FFFC0C14 | 8, 16          |
| PFC            | Port A I/O register                 | PAIOR        | 16             | H'FFFE3806 | 8, 16          |
|                | Port A control register 4           | PACR4        | 16             | H'FFFE3810 | 8, 16, 32      |
|                | Port A control register 3           | PACR3        | 16             | H'FFFE3812 | 8, 16          |
|                | Port A control register 2           | PACR2        | 16             | H'FFFE3814 | 8, 16, 32      |
|                | Port A control register 1           | PACR1        | 16             | H'FFFE3816 | 8, 16          |
|                | Port B I/O register                 | PBIOR        | 16             | H'FFFE3886 | 8, 16          |
|                | Port B control register 4           | PBCR4        | 16             | H'FFFE3890 | 8, 16, 32      |
|                | Port B control register 3           | PBCR3        | 16             | H'FFFE3892 | 8, 16          |
|                | Port B control register 2           | PBCR2        | 16             | H'FFFE3894 | 8, 16, 32      |
|                | Port B control register 1           | PBCR1        | 16             | H'FFFE3896 | 8, 16          |
|                | Port C I/O register                 | PCIOR        | 16             | H'FFFE3906 | 8, 16          |
|                | Port C control register 4           | PCCR4        | 16             | H'FFFE3910 | 8, 16, 32      |
|                | Port C control register 3           | PCCR3        | 16             | H'FFFE3912 | 8, 16          |

| Module   | Register Name             | Abbreviation | Number of Bits | Address    | Access<br>Size |
|----------|---------------------------|--------------|----------------|------------|----------------|
| PFC      | Port C control register 2 | PCCR2        | 16             | H'FFFE3914 | 8, 16, 32      |
|          | Port C control register 1 | PCCR1        | 16             | H'FFFE3916 | 8, 16          |
|          | Port D I/O register       | PDIOR        | 16             | H'FFFFC808 | 8, 16          |
|          | Port D control register 2 | PDCR2        | 16             | H'FFFFC80C | 8, 16, 32      |
|          | Port D control register 1 | PDCR1        | 16             | H'FFFFC80E | 8, 16          |
|          | Port E I/O register       | PEIOR        | 16             | H'FFFFC818 | 8, 16          |
|          | Port E control register 2 | PECR2        | 16             | H'FFFFC81C | 8, 16, 32      |
|          | Port E control register 1 | PECR1        | 16             | H'FFFFC81E | 8, 16          |
|          | Port F I/O register       | PFIOR        | 16             | H'FFFFC82A | 8, 16          |
|          | Port F control register 2 | PFCR2        | 16             | H'FFFFC82C | 8, 16, 32      |
|          | Port F control register 1 | PFCR1        | 16             | H'FFFFC82E | 8, 16          |
|          | Port G I/O register       | PGIOR        | 16             | H'FFFFC83C | 8, 16          |
|          | Port G control register 2 | PGCR2        | 16             | H'FFFFC840 | 8, 16, 32      |
|          | Port G control register 1 | PGCR1        | 16             | H'FFFFC842 | 8, 16          |
|          | Port H I/O register       | PHIOR        | 16             | H'FFFFC854 | 8, 16          |
|          | Port H control register   | PHCR         | 16             | H'FFFFC858 | 8, 16          |
|          | Port J I/O register       | PJIOR        | 16             | H'FFFFC86C | 8, 16          |
|          | Port J control register 2 | PJCR2        | 16             | H'FFFFC870 | 8, 16, 32      |
|          | Port J control register 1 | PJCR1        | 16             | H'FFFFC872 | 8, 16          |
|          | Port K I/O register       | PKIOR        | 16             | H'FFFFC88C | 8, 16          |
|          | Port K control register 2 | PKCR2        | 16             | H'FFFFC890 | 8, 16, 32      |
|          | Port K control register 1 | PKCR1        | 16             | H'FFFFC892 | 8, 16          |
|          | Port L I/O register       | PLIOR        | 16             | H'FFFFC8A8 | 8, 16          |
|          | Port L control register 2 | PLCR2        | 16             | H'FFFFC8AC | 8, 16, 32      |
|          | Port L control register 1 | PLCR1        | 16             | H'FFFFC8AE | 8, 16          |
| I/O port | Port A data register      | PADR         | 16             | H'FFFE3802 | 8, 16          |
|          | Port A port register      | PAPR         | 16             | H'FFFE381E | 8, 16          |
|          | Port B data register      | PBDR         | 16             | H'FFFE3882 | 8, 16          |
|          | Port B port register      | PBPR         | 16             | H'FFFE389E | 8, 16          |
|          | Port B inverting register | PBIR         | 16             | H'FFFE3898 | 8, 16, 32      |

| Module   | Register Name                           | Abbreviation | Number of Bits | Address    | Access<br>Size |
|----------|---|--------------|----------------|------------|----------------|
| I/O port | Port B driving ability setting register | PBDSR        | 16             | H'FFFE389A | 8, 16          |
|          | Port B pin state setting register       | PBPSR        | 16             | H'FFFE389C | 8, 16, 32      |
|          | Port C data register                    | PCDR         | 16             | H'FFFE3902 | 8, 16          |
|          | Port C port register                    | PCPR         | 16             | H'FFFE391E | 8, 16          |
|          | Port D data register                    | PDDR         | 16             | H'FFFFC800 | 8, 16, 32      |
|          | Port D port register                    | PDPR         | 16             | H'FFFFC802 | 8, 16          |
|          | Port D inverting register               | PDIR         | 16             | H'FFFFC804 | 8, 16          |
|          | Port E data register                    | PEDR         | 16             | H'FFFFC810 | 8, 16, 32      |
|          | Port E port register                    | PEPR         | 16             | H'FFFFC812 | 8, 16          |
|          | Port E inverting register               | PEIR         | 16             | H'FFFFC814 | 8, 16, 32      |
|          | Port E driving ability setting register | PEDSR        | 16             | H'FFFFC816 | 8, 16          |
|          | Port F data register PFDR 1             | 16           | H'FFFFC820     | 8, 16, 32  |                |
|          | Port F port register                    | PFPR         | 16             | H'FFFFC822 | 8, 16          |
|          | Port F inverting register               | PFIR         | 16             | H'FFFFC824 | 8, 16, 32      |
|          | Port F driving ability setting register | PFDSR        | 16             | H'FFFFC826 | 8, 16          |
|          | Port F pin state setting register       | PFPSR        | 16             | H'FFFFC828 | 8, 16, 32      |
|          | Port G data register                    | PGDR         | 16             | H'FFFFC830 | 8, 16, 32      |
|          | Port G port register                    | PGPR         | 16             | H'FFFFC832 | 8, 16          |
|          | Port G inverting register               | PGIR         | 16             | H'FFFFC834 | 8, 16, 32      |
|          | Port G driving ability setting register | PGDSR        | 16             | H'FFFFC836 | 8, 16          |
|          | Port G edge selecting register          | PGER         | 16             | H'FFFFC838 | 8, 16          |
|          | Port H data register                    | PHDR         | 16             | H'FFFFC850 | 8, 16, 32      |
|          | Port H port register                    | PHPR         | 16             | H'FFFFC852 | 8, 16          |
|          | Port J data register                    | PJDR         | 16             | H'FFFFC860 | 8, 16, 32      |
|          | Port J port register                    | PJPR         | 16             | H'FFFFC862 | 8, 16          |
|          | Port J inverting register               | PJIR         | 16             | H'FFFFC864 | 8, 16, 32      |
|          | Port J driving ability setting register | PJDSR        | 16             | H'FFFFC866 | 8, 16          |

| Module   | Register Name                                | Abbreviation | Number of Bits | Address    | Access<br>Size |
|----------|--|--------------|----------------|------------|----------------|
| I/O port | Port J pin state setting register            | PJPSR        | 16             | H'FFFFC868 | 8, 16          |
|          | Port K data register                         | PKDR         | 16             | H'FFFFC880 | 8, 16, 32      |
|          | Port K port register                         | PKPR         | 16             | H'FFFFC882 | 8, 16          |
|          | Port K inverting register                    | PKIR         | 16             | H'FFFFC884 | 8, 16, 32      |
|          | Port K driving ability setting register      | PKDSR        | 16             | H'FFFFC886 | 8, 16          |
|          | Port K pin state setting register            | PKPSR        | 16             | H'FFFFC888 | 8, 16          |
|          | Port L data register                         | PLDR         | 16             | H'FFFFC8A0 | 8, 16, 32      |
|          | Port L port register                         | PLPR         | 16             | H'FFFFC8A2 | 8, 16          |
|          | Port L inverting register                    | PLIR         | 16             | H'FFFFC8A4 | 8, 16          |
|          | CK control register                          | CKCR         | 16             | H'FFFFC920 | 8, 16          |
| MISG     | Calculation data register                    | MISRCDR      | 32 H'FFF7FFC   | 8, 16, 32  |                |
|          | MISR control register                        | MISRCR       | 8              | H'FFFC1C00 | 8              |
|          | Multi-input signature register               | MISR         | 32             | H'FFFC1C04 | 32             |
| ROM/     | Flash pin monitor register                   | FPMON        | 8              | H'FFFFA800 | 8              |
| EEPROM   | Flash mode register                          | FMODR        | 8              | H'FFFFA802 | 8              |
|          | Flash access status register                 | FASTAT       | 8              | H'FFFFA810 | 8              |
|          | Flash access error interrupt enable register | FAEINT       | 8              | H'FFFFA811 | 8              |
|          | ROM MAT select register                      | ROMMAT       | 16             | H'FFFFA820 | 8, 16          |
|          | FCU RAM enable register                      | FCURAME      | 16             | H'FFFFA854 | 8, 16          |
|          | Flash status register 0                      | FSTATR0      | 8              | H'FFFFA900 | 8, 16          |
|          | Flash status register 1                      | FSTATR1      | 8              | H'FFFFA901 | 8, 16          |
|          | Flash P/E mode entry register                | FENTRYR      | 16             | H'FFFFA902 | 8, 16          |
|          | Flash protect register                       | FPROTR       | 16             | H'FFFFA904 | 8, 16          |
|          | Flash reset register                         | FRESETR      | 16             | H'FFFFA906 | 8, 16          |
|          | FCU command register                         | FCMDR        | 16             | H'FFFFA90A | 8, 16          |
|          | FCU RAM ECC error control register           | FRAMECCR     | 8              | H'FFFFA90C | 8              |
|          | FCU processing switch register               | FCPSR        | 16             | H'FFFFA918 | 8, 16          |
|          | EEPROM blank check control register          | EEPBCCNT     | 16             | H'FFFFA91A | 8, 16          |

| Module                 | Register Name                          | Abbreviation | Number of Bits | Address    | Access<br>Size |
|------------------------|--|--------------|----------------|------------|----------------|
| ROM/                   | Flash P/E status register              | FPESTAT      | 16             | H'FFFFA91C | 8, 16          |
| EEPROM                 | EEPROM blank check status register     | EEPBCSTAT    | 16             | H'FFFFA91E | 8, 16          |
|                        | EEPROM read enable register 0          | EEPRE0       | 16             | H'FFFFA840 | 8, 16          |
|                        | EEPROM read enable register 1          | EEPRE1       | 16             | H'FFFFA842 | 8, 16          |
|                        | EEPROM program/erase enable register 0 | EEPWE0       | 16             | H'FFFFA850 | 8, 16          |
|                        | EEPROM program/erase enable register 1 | EEPWE1       | 16             | H'FFFFA852 | 8, 16          |
|                        | EEPROM MAT select register             | EEPMAT       | 16             | H'FFFFAB00 | 8, 16          |
| ROMC                   |  | H'FFFC1400   | 32             |            |                |
|                        | ROM cache control register 2           | RCCR2        | 32             | H'FFFC1408 | 32             |
| RAM                    | RAM enable control register 0          | RAMEN0       | 16             | H'FFFF0800 | 8, (16)        |
|                        | RAM write enable control register 0    | RAMWEN0      | 16             | H'FFFF0802 | 8, (16)        |
|                        | RAM enable control register 1          | RAMEN1       | 16             | H'FFFF0840 | 8, (16)        |
|                        | RAM write enable control register 1    | RAMWEN1      | 16             | H'FFFF0842 | 8, (16)        |
|                        | RAM ECC enable control register        | RAMECC       | 16             | H'FFFF0804 | 8, (16)        |
|                        | RAM error status register              | RAMERR       | 8              | H'FFFF0806 | 8              |
|                        | RAM error interrupt control register   | RAMINT       | 8              | H'FFFF0810 | 8              |
|                        | RAM access cycle set register          | RAMACYC      | 16             | H'FFFF0812 | 8, 16          |
| Power-<br>down<br>mode | Standby control register               | STBCR        | 16             | H'FFFE0400 | 8, 16          |

## 31.2 List of Register Bits

The addresses and bit names of the registers in the peripheral modules are shown below. Registers composed of 16 and 32 bits are divided into 2 and 4 rows, accordingly.

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|------------|-----------|
| INTC   | ICR0             | NMIL       | _          | _          | _          |            | _          | _          | NMIE      |
|        |                  | _          | _          | _          | _          |            | _          | _          | _         |
|        | ICR1             | IRQ71S     | IRQ70S     | IRQ61S     | IRQ60S     | IRQ51S     | IRQ50S     | IRQ41S     | IRQ40S    |
|        |                  | IRQ31S     | IRQ30S     | IRQ21S     | IRQ20S     | IRQ11S     | IRQ10S     | IRQ01S     | IRQ00S    |
|        | IRQRR            |            | _          | _          | _          | _          | _          | _          | _         |
|        |                  | IRQ7F      | IRQ6F      | IRQ5F      | IRQ4F      | IRQ3F      | IRQ2F      | IRQ1F      | IRQ0F     |
|        | IBCR             | E15        | E14        | E13        | E12        | E11        | E10        | E9         | E8        |
|        |                  | E7         | E6         | E5         | E4         | E3         | E2         | E1         | _         |
|        | IBNR             | BE         | [1:0]      | BOVE       | _          | _          | _          | _          |           |
|        |                  | _          | _          | _          | _          | BN[3:0]    |            |            |           |
|        | SINTR1           | _          | _          | _          | _          | _          | _          | _          | SINTC     |
|        | SINTR2           | _          | _          | _          | _          | _          | _          | _          | SINTC     |
|        | SINTR3           | _          | _          | _          | _          | _          | _          | _          | SINTC     |
|        | SINTR4           | _          | _          | _          | _          | _          | _          | _          | SINTC     |
|        | SINTR5           |            | _          | _          | _          | _          | _          | _          | SINTC     |
|        | SINTR6           |            | _          | _          | _          |            | _          | _          | SINTC     |
|        | SINTR7           |            | _          | _          | _          | _          | _          | _          | SINTC     |
|        | SINTR8           |            | _          | _          | _          | _          | _          | _          | SINTC     |
|        | IPR01            |            | IR         | Q0         |            |            | IRO        | 21         | •         |
|        |                  |            | IR         | Q2         |            |            | IRO        | 23         |           |
|        | IPR02            |            | IR         | Q4         |            |            | IRO        | <b>Q</b> 5 |           |
|        |                  |            | IR         | Q6         |            |            | IRO        | <b>Q</b> 7 |           |
|        | SINTR9           | _          | _          |            |            | _          |            | _          | SINTC     |
|        | SINTR10          | _          |            |            |            |            |            |            | SINTC     |
|        | SINTR11          | _          | _          | _          | _          | _          | _          | _          | SINTC     |
|        | SINTR12          | _          | _          | _          | _          | _          | _          | _          | SINTC     |
|        | SINTR13          | _          |            |            |            | _          |            | _          | SINTC     |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6  | 29/21/13/5   | 28/20/12/4 | 27/19/11/3                | 26/18/10/2  | 25/17/9/1    | 24/16/8/0 |  |
|--------|------------------|------------|-------------|--------------|------------|---------------------------|-------------|--------------|-----------|--|
| INTC   | SINTR14          |            |             |              |            |                           |             | _            | SINTC     |  |
|        | SINTR15          |            | _           |              | _          |                           |             | _            | SINTC     |  |
|        | IPR03            |            | DM          | AC0          |            |                           | DM          | AC1          | 11        |  |
|        |                  |            | DM          | AC2          |            | DMAC3                     |             |              |           |  |
|        | IPR04            |            | DM          | AC4          |            | DMAC5                     |             |              |           |  |
|        |                  |            | DM          | AC6          |            |                           | DM          | AC7          |           |  |
|        | IPR05            |            | CV          | <b>ИТО</b>   |            |                           | CM          | IT1          |           |  |
|        |                  |            | Res         | erved        |            |                           | WI          | DT           |           |  |
|        | IPR06            |            | ATU-A (IC   | IA0, ICIA1)  |            |                           | ATU-A (ICI  | IA2, ICIA3)  |           |  |
|        |                  |            | ATU-A (IC   | IA4, ICIA5)  |            |                           | ATU-A       | (OVIA)       |           |  |
|        | IPR07            |            | ATU-B (CM   | IB0, CMIB1)  |            |                           | ATU-B (CM   | IIB6, ICIB0) |           |  |
|        |                  | А          | TU-C0 (IMIC | 000 to IMICO | 3)         | ATU-C0 (OVIC0)            |             |              |           |  |
|        | IPR08            | А          | TU-C1 (IMIC | C10 to IMIC1 | 3)         | ATU-C1 (OVIC1)            |             |              |           |  |
|        |                  | А          | TU-C2 (IMIC | C20 to IMIC2 | 3)         | ATU-C2 (OVIC2)            |             |              |           |  |
|        | IPR09            | А          | TU-C3 (IMIC | C30 to IMIC3 | 3)         |                           | ATU-C3      | (OVIC3)      |           |  |
|        |                  | А          | TU-C4 (IMIC | C40 to IMIC4 | 3)         |                           | ATU-C4      | (OVIC4)      |           |  |
|        | IPR10            | ΑΤ         | TU-D0 (CMIE | 000 to CMID  | 03)        | ATU-D0 (OVI1D0, OVI2D0)   |             |              |           |  |
|        |                  | ΑT         | TU-D0 (UDIE | 000 to UDID  | 03)        | ATU-D1 (CMID10 to CMID13) |             |              |           |  |
|        | IPR11            | А          | TU-D1 (OVI  | 1D1, OVI2D   | 1)         | ATU-D1 (UDID10 to UDID13) |             |              |           |  |
|        |                  | ΓA         | U-D2 (CMIE  | 020 to CMID  | 23)        | ATU-D2 (OVI1D2, OVI2D2)   |             |              |           |  |
|        | IPR12            | A          | TU-D2 (UDIE | 020 to UDID  | 23)        | AT                        | U-D3 (CMID  | 30 to CMID   | 33)       |  |
|        |                  | А          | TU-D3 (OVI  | 1D3, OVI2D   | 3)         | ATU-D3 (UDID30 to UDID33) |             |              |           |  |
|        | IPR13            |            | Rese        | erved        |            |                           | Rese        | erved        |           |  |
|        |                  |            | Rese        | erved        |            |                           | Rese        | erved        |           |  |
|        | IPR14            |            | Rese        | erved        |            |                           | Rese        | erved        |           |  |
|        |                  | Α٦         | ΓU-E0 (CMIE | E00 to CMIE  | 03)        | AT                        | U-E1 (CMIE  | 10 to CMIE   | 13)       |  |
|        | IPR15            | A          | ΓU-E2 (CMIE | E20 to CMIE  | 23)        | AT                        | U-E3 (CMIE  | 30 to CMIE   | 33)       |  |
|        |                  | Α٦         | ΓU-E4 (CMIE | E40 to CMIE  | 43)        | ATU-E5 (CMIE50 to CMIE53) |             |              |           |  |
|        | IPR16            |            | ATU-F (ICI  | F0 to ICIF3) |            |                           | ATU-F (ICIF | F4 to ICIF7) |           |  |
|        |                  |            | ATU-F (ICIF | 8 to ICIF11) |            | ,                         | ATU-F (ICIF | 12 to ICIF15 | )         |  |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6  | 29/21/13/5   | 28/20/12/4 | 27/19/11/3               | 26/18/10/2  | 25/17/9/1   | 24/16/8/0 |  |
|--------|------------------|------------|-------------|--------------|------------|--------------------------|-------------|-------------|-----------|--|
| INTC   | IPR17            |            | ATU-F (ICIF | 16 to ICIF19 | )          | Reserved                 |             |             |           |  |
|        |                  |            | Res         | erved        |            | Reserved                 |             |             |           |  |
|        | IPR18            |            | ATU-F (OVI  | F0 to OVIF3  | )          | ATU-F (OVIF4 to OVIF7)   |             |             |           |  |
|        |                  | ,          | ATU-F (OVI  | 8 to OVIF1   | 1)         | ATU-F (OVIF12 to OVIF15) |             |             |           |  |
|        | IPR19            | A          | TU-F (OVIF  | 16 to OVIF1  | 9)         | Reserved                 |             |             |           |  |
|        |                  |            | Res         | erved        |            | Reserved                 |             |             |           |  |
|        | IPR20            | ,          | ATU-G (CMI  | G0 to CMIG   | 3)         |                          | ATU-G (CMI  | G4, CMIG5)  | )         |  |
|        |                  |            | ATU-H       | (CMIH)       |            |                          | Rese        | erved       |           |  |
|        | IPR21            |            | ATU-J (DF   | IJ0, DFIJ1)  |            |                          | ATU-J (OV   | IJ0, OVIJ1) |           |  |
|        |                  |            | ATU-J (DOV  | 'IJ0, DOVIJ1 | )          |                          | Rese        | erved       |           |  |
|        | IPR22            |            | ADC         | (ADI0)       |            | ADC (ADI1)               |             |             |           |  |
|        |                  |            | ADC (ADID   | 00 to ADID3) |            | ADC (ADID4 to ADID7)     |             |             |           |  |
|        | IPR23            |            | ADC (ADID   | 8 to ADID11  | )          | A                        | ADC (ADID1: | 2 to ADID15 | )         |  |
|        |                  |            | ADC (A      | ADID40)      |            |                          | ADC (A      | DID41)      |           |  |
|        | IPR24            |            | ADC (A      | ADID42)      |            |                          | ADC (A      | DID43)      |           |  |
|        |                  |            | ADC (A      | ADID44)      |            |                          | ADC (A      | DID45)      |           |  |
|        | IPR25            |            | ADC (A      | ADID46)      |            | ADC (ADID47)             |             |             |           |  |
|        |                  |            | Res         | erved        |            | Reserved                 |             |             |           |  |
|        | IPR26            |            | sc          | I_A          |            | SCI_B                    |             |             |           |  |
|        |                  |            | SC          | CI_C         |            | SCI_D                    |             |             |           |  |
|        | IPR27            |            | SC          | I_E          |            | RSPI_A                   |             |             |           |  |
|        |                  |            | RS          | PI_B         |            | RSPI_C                   |             |             |           |  |
|        | IPR28            |            | RCA         | AN_A         |            |                          | RCA         | N_B         |           |  |
|        |                  |            | RCA         | AN_C         |            |                          | Rese        | erved       |           |  |
|        | IPR29            |            | A-D         | MAC          |            |                          | Rese        | erved       |           |  |
|        |                  |            | Res         | erved        |            |                          | Rese        | erved       |           |  |
| UBC    | BAR_0            | BA0_31     | BA0_30      | BA0_29       | BA0_28     | BA0_27                   | BA0_26      | BA0_25      | BA0_24    |  |
|        |                  | BA0_23     | BA0_22      | BA0_21       | BA0_20     | BA0_19                   | BA0_18      | BA0_17      | BA0_16    |  |
|        |                  | BA0_15     | BA0_14      | BA0_13       | BA0_12     | BA0_11                   | BA0_10      | BA0_9       | BA0_8     |  |
|        |                  | BA0_7      | BA0_6       | BA0_5        | BA0_4      | BA0_3                    | BA0_2       | BA0_1       | BA0_0     |  |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| UBC    | BAMR_0           | BAM0_31    | BAM0_30    | BAM0_29    | BAM0_28    | BAM0_27    | BAM0_26    | BAM0_25   | BAM0_24   |
|        |                  | BAM0_23    | BAM0_22    | BAM0_21    | BAM0_20    | BAM0_19    | BAM0_18    | BAM0_17   | BAM0_16   |
|        |                  | BAM0_15    | BAM0_14    | BAM0_13    | BAM0_12    | BAM0_11    | BAM0_10    | BAM0_9    | BAM0_8    |
|        |                  | BAM0_7     | BAM0_6     | BAM0_5     | BAM0_4     | BAM0_3     | BAM0_2     | BAM0_1    | BAM0_0    |
|        | BBR_0            | _          | _          | UBID0      | _          |            |            | CP0_[2:0] | ı         |
|        |                  | CD0        | _[1:0]     | ID0_       | _[1:0]     | RW0        | _[1:0]     | SZ0_      | _[1:0]    |
|        | BAR_1            | BA1_31     | BA1_30     | BA1_29     | BA1_28     | BA1_27     | BA1_26     | BA1_25    | BA1_24    |
|        |                  | BA1_23     | BA1_22     | BA1_21     | BA1_20     | BA1_19     | BA1_18     | BA1_17    | BA1_16    |
|        |                  | BA1_15     | BA1_14     | BA1_13     | BA1_12     | BA1_11     | BA1_10     | BA1_9     | BA1_8     |
|        |                  | BA1_7      | BA1_6      | BA1_5      | BA1_4      | BA1_3      | BA1_2      | BA1_1     | BA1_0     |
|        | BAMR_1           | BAM1_31    | BAM1_30    | BAM1_29    | BAM1_28    | BAM1_27    | BAM1_26    | BAM1_25   | BAM1_24   |
|        |                  | BAM1_23    | BAM1_22    | BAM1_21    | BAM1_20    | BAM1_19    | BAM1_18    | BAM1_17   | BAM1_16   |
|        |                  | BAM1_15    | BAM1_14    | BAM1_13    | BAM1_12    | BAM1_11    | BAM1_10    | BAM1_9    | BAM1_8    |
|        |                  | BAM1_7     | BAM1_6     | BAM1_5     | BAM1_4     | BAM1_3     | BAM1_2     | BAM1_1    | BAM1_0    |
|        | BBR_1            | _          | _          | UBID1      | _          | _          | CP1_[2:0]  |           |           |
|        |                  | CD1_[1:0]  |            | ID1_[1:0]  |            | RW1        | _[1:0]     | SZ1_      | _[1:0]    |
|        | BAR_2            | BA2_31     | BA2_30     | BA2_29     | BA2_28     | BA2_27     | BA2_26     | BA2_25    | BA2_24    |
|        |                  | BA2_23     | BA2_22     | BA2_21     | BA2_20     | BA2_19     | BA2_18     | BA2_17    | BA2_16    |
|        |                  | BA2_15     | BA2_14     | BA2_13     | BA2_12     | BA2_11     | BA2_10     | BA2_9     | BA2_8     |
|        |                  | BA2_7      | BA2_6      | BA2_5      | BA2_4      | BA2_3      | BA2_2      | BA2_1     | BA2_0     |
|        | BAMR_2           | BAM2_31    | BAM2_30    | BAM2_29    | BAM2_28    | BAM2_27    | BAM2_26    | BAM2_25   | BAM2_24   |
|        |                  | BAM2_23    | BAM2_22    | BAM2_21    | BAM2_20    | BAM2_19    | BAM2_18    | BAM2_17   | BAM2_16   |
|        |                  | BAM2_15    | BAM2_14    | BAM2_13    | BAM2_12    | BAM2_11    | BAM2_10    | BAM2_9    | BAM2_8    |
|        |                  | BAM2_7     | BAM2_6     | BAM2_5     | BAM2_4     | BAM2_3     | BAM2_2     | BAM2_1    | BAM2_0    |
|        | BBR_2            | _          | _          | UBID2      | _          | _          |            | CP2_[2:0] | ı         |
|        |                  | CD2        | _[1:0]     | ID2_       | _[1:0]     | RW2        | _[1:0]     | SZ2_      | _[1:0]    |
|        | BAR_3            | BA3_31     | BA3_30     | BA3_29     | BA3_28     | BA3_27     | BA3_26     | BA3_25    | BA3_24    |
|        |                  | BA3_23     | BA3_22     | BA3_21     | BA3_20     | BA3_19     | BA3_18     | BA3_17    | BA3_16    |
|        |                  | BA3_15     | BA3_14     | BA3_13     | BA3_12     | BA3_11     | BA3_10     | BA3_9     | BA3_8     |
|        |                  | BA3_7      | BA3_6      | BA3_5      | BA3_4      | BA3_3      | BA3_2      | BA3_1     | BA3_0     |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|------------|-----------|
| UBC    | BAMR_3           | BAM3_31    | BAM3_30    | BAM3_29    | BAM3_28    | BAM3_27    | BAM3_26    | BAM3_25    | BAM3_24   |
|        |                  | BAM3_23    | BAM3_22    | BAM3_21    | BAM3_20    | BAM3_19    | BAM3_18    | BAM3_17    | BAM3_16   |
|        |                  | BAM3_15    | BAM3_14    | BAM3_13    | BAM3_12    | BAM3_11    | BAM3_10    | BAM3_9     | BAM3_8    |
|        |                  | BAM3_7     | BAM3_6     | BAM3_5     | BAM3_4     | BAM3_3     | BAM3_2     | BAM3_1     | BAM3_0    |
|        | BBR_3            | _          | _          | UBID3      | _          | _          |            | CP3_[2:0]  |           |
|        |                  | CD3        | _[1:0]     | ID3_       | _[1:0]     | RW3        | _[1:0]     | SZ3_       | _[1:0]    |
|        | BRCR             | _          | _          | _          | _          | _          | _          | _          |           |
|        |                  | _          | _          | UTOD3      | UTOD2      | UTOD1      | UTOD0      | CKS        | [1:0]     |
|        |                  | SCMFC0     | SCMFC1     | SCMFC2     | SCMFC3     | SCMFD0     | SCMFD1     | SCMFD2     | SCMFD3    |
|        |                  | РСВ3       | PCB2       | PCB1       | PCB0       | _          | _          | _          | _         |
| BSC    | CS0BCR           | _          |            | IWW[2:0]   |            |            | IWRWD[2:0  | ]          | IWRWS[2]  |
|        |                  | IWRW       | VS[1:0]    |            | IWRRD[2:0] |            |            | IWRRS[2:0] |           |
|        |                  | _          | _          | TYP        | E[1:0]     |            | BSZ        | [1:0]      |           |
|        |                  | _          | _          | _          | _          | _          | _          |            |           |
|        | CS1BCR           | CR —       |            | IWW[2:0]   |            |            | IWRWD[2:0  | ]          | IWRWS[2]  |
|        |                  | IWRW       | VS[1:0]    |            | IWRRD[2:0] |            |            | IWRRS[2:0] |           |
|        |                  |            | _          | TYP        | E[1:0]     | _          | BSZ        | [1:0]      | _         |
|        |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|        | CS2BCR           |            |            | IWW[2:0]   |            |            | IWRWD[2:0  | ]          | IWRWS[2]  |
|        |                  | IWRW       | VS[1:0]    |            | IWRRD[2:0] |            |            | IWRRS[2:0] |           |
|        |                  | _          | _          | TYP        | E[1:0]     |            | BSZ        | [1:0]      |           |
|        |                  | _          | _          | _          | _          | _          | _          |            |           |
|        | CS3BCR           | _          |            | IWW[2:0]   |            |            | IWRWD[2:0  | ]          | IWRWS[2]  |
|        |                  | IWRW       | VS[1:0]    |            | IWRRD[2:0] |            |            | IWRRS[2:0] |           |
|        |                  | _          | _          | TYP        | E[1:0]     | _          | BSZ        | [1:0]      | _         |
|        |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|        | CS0WCR           |            | _          | _          | _          |            |            |            | _         |
|        |                  | _          | _          | _          | BAS        |            |            | WW[2:0]    |           |
|        |                  | _          | _          | _          | sw         | [1:0]      |            | WR[3:1]    |           |
|        |                  | WR[0]      | WM         | _          | _          |            | _          | HW         | [1:0]     |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| BSC    | CS1WCR           | _          | _          | _          | _          | _          |            | _         | _         |
|        |                  | _          | _          | _          | BAS        | _          |            | WW[2:0]   | •         |
|        |                  | _          | _          | _          | SW         | [1:0]      |            | WR[3:1]   |           |
|        |                  | WR[0]      | WM         | _          | _          | _          | _          | HW[       | [1:0]     |
|        | CS2WCR           | -          | _          | _          | _          | _          | _          | _         | _         |
|        |                  | -          |            | _          | BAS        | _          |            | WW[2:0]   |           |
|        |                  | _          | _          | _          | SW         | [1:0]      |            | WR[3:1]   |           |
|        |                  | WR[0]      | WM         | _          | _          |            | _          | HW[       | [1:0]     |
|        | CS3WCR           |            | _          |            |            |            |            | _         | _         |
|        |                  | _          | _          | _          | BAS        | _          |            | WW[2:0]   |           |
|        |                  | _          | _          | _          | SW         | [1:0]      |            | WR[3:1]   |           |
|        |                  | WR[0]      | WM         | _          | _          | _          |            | HW[       | [1:0]     |
| DMAC   | SAR0             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DAR0             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DMATCR0          | _          | _          | _          | _          |            | _          |           | _         |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | CHCR0            | TC1        | _          | _          | RLD1       | _          | _          | _         |           |
|        |                  | _          |            | _          |            | _          | HIE        |           |           |
|        |                  | DM         | [1:0]      | SM         | [1:0]      |            | RS[        | 3:0]      |           |
|        |                  | _          | _          | ТВ         | TS[        | 1:0]       | IE         | _         | DE        |
|        | CHFR0            | _          | <u> </u>   | _          | HE         | _          | _          | _         | TE        |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| DMAC   | TEMSK0           |            |            |            | TEMKE      |            |            |           |           |
|        |                  | _          | _          | _          | _          |            | _          |           | TEMASK    |
|        | RSAR0            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RDAR0            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RDMATCR0         | _          |            |            |            |            |            | _         | _         |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | SAR1             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DAR1             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DMATCR1          | _          | _          | _          | _          | _          | _          | _         | _         |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | CHCR1            | TC1        | _          | _          | RLD1       | _          |            | _         | _         |
|        |                  | _          | <u> </u>   | _          | <u> -</u>  | _          | HIE        | <u> </u>  | _         |
|        |                  | DM         | [1:0]      |            | [1:0]      | 1          |            | S[3:0]    |           |
|        |                  | _          | _          | ТВ         |            | [1:0]      | IE         |           | DE        |
|        | CHFR1            |            |            |            | HE         |            |            |           | TE        |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1   | 24/16/8/0  |
|--------|------------------|------------|------------|------------|------------|------------|------------|-------------|------------|
| DMAC   | TEMSK1           | 0172071077 | 00/22/14/0 | 20/21/10/0 | TEMKE      |            | 20/10/10/2 | 20, 11,0, 1 | 2-1/10/0/0 |
| DIVIAO | TEMORT           |            |            | <u> </u>   |            |            | <u> </u>   | <u></u>     | TEMASK     |
|        | RSAR1            |            |            |            |            |            |            |             | TEMACK     |
|        | 11071111         |            |            |            |            |            |            |             | 1          |
|        |                  |            |            |            |            |            |            |             | 1          |
|        |                  |            |            |            |            |            |            |             |            |
|        | RDAR1            |            |            |            |            |            |            |             |            |
|        | HUAITI           |            |            |            |            |            |            |             |            |
|        |                  |            |            |            |            |            |            |             | 1          |
|        |                  |            |            |            |            |            |            |             | 1          |
|        | RDMATCR1         |            |            |            |            |            |            |             | 1          |
|        | NDWATORT         |            |            |            |            |            |            |             |            |
|        |                  |            |            |            |            |            |            |             |            |
|        |                  |            |            |            |            |            |            |             | 1          |
|        | SAR2             |            |            |            |            |            |            |             | <u> </u>   |
|        | SAR2             |            |            |            |            |            |            |             |            |
|        |                  |            |            |            |            |            |            |             | <u> </u>   |
|        |                  |            |            |            |            |            |            |             |            |
|        | D. I. D. C.      |            |            |            |            |            |            |             | 1          |
|        | DAR2             |            |            |            |            |            |            |             | -          |
|        |                  |            |            |            |            |            |            |             |            |
|        |                  |            |            |            |            |            |            |             |            |
|        |                  |            |            |            |            |            |            |             |            |
|        | DMATCR2          |            |            | _          | _          | _          | _          | _           | _          |
|        |                  |            |            |            |            |            |            |             |            |
|        |                  |            |            |            |            |            |            |             | 1          |
|        |                  |            |            |            |            |            |            |             |            |
|        | CHCR2            | TC1        | _          |            | RLD1       |            |            | _           | _          |
|        |                  | _          | _          | _          | _          | _          | HIE        | _           | _          |
|        |                  | DM         | [1:0]      |            | [1:0]      |            |            | [3:0]       |            |
|        |                  |            | _          | ТВ         | TS[        | 1:0]       | IE         | _           | DE         |
|        | CHFR2            | _          | _          | _          | HE         | _          | _          | _           | TE         |

RENESAS

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| DMAC   | TEMSK2           |            | 1          | <u>I</u>   | TEMKE      |            | <u>I</u>   |           |           |
|        |                  |            | _          | _          | _          | _          | _          | _         | TEMASK    |
|        | RSAR2            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RDAR2            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RDMATCR2         | _          | _          | _          | _          | _          |            |           | _         |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | SAR3             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DAR3             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DMATCR3          |            | _          | _          | _          | _          | _          |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | CHCR3            | TC1        |            | _          | RLD1       | _          | _          | _         | _         |
|        |                  |            | _          | _          |            | _          | HIE        | _         | _         |
|        |                  | DM         | [1:0]      | SM[1:0]    |            | 1          |            | [3:0]     |           |
|        |                  | _          |            | ТВ         |            | [1:0]      | IE         | _         | DE        |
|        | CHFR3            |            |            |            | HE         |            |            |           | TE        |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2    | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|---------------|-----------|-----------|
| DMAC   | TEMSK3           |            |            | ]          | TEMKE      |            |               |           |           |
|        |                  |            | _          | _          | _          | _          | _             |           | TEMASK    |
|        | RSAR3            |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | RDAR3            |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | RDMATCR3         |            |            | _          |            |            |               | _         | _         |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | SAR4             |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | DAR4             |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | DMATCR4          |            | _          | _          |            | _          | _             | _         | <u> </u>  |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | OLIOD4           | TO         | 1.01       |            | DI DI      | DI D       | 014 03        |           | .ET       |
|        | CHCR4            | 10         | [1:0]      |            | RLD1       | HLD:       | 2[1:0]<br>HIE |           | IFT       |
|        |                  |            | [1:0]      | CNA        | [1:0]      |            |               | 2.01      | <u> </u>  |
|        |                  | DIM        | [1:0]      | ТВ         | [1:0]      | [1:0] IE   |               | 3:0]      | DE        |
|        | CHFR4            |            |            | I D        | HE         | 1.0]       | IC .          |           | TE        |
|        | CHFH4            |            |            |            | ПЕ         |            |               | _         | i E       |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| DMAC   | TEMSK4           |            | •          |            | TEMKE      | Y[7:0]     |            | •         | •         |
|        |                  | _          | _          |            |            | _          |            | _         | TEMASK    |
|        | RSAR4            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RDAR4            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RDMATCR4         |            |            |            |            |            |            |           | _         |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | 10001            |            |            |            |            |            |            |           |           |
|        | ARCR4            | _          |            |            |            |            |            | _         |           |
|        | RARCR4           |            |            |            |            |            |            |           |           |
|        | HARCH4           |            |            |            |            |            |            |           |           |
|        | SAR5             |            |            |            |            |            |            |           |           |
|        | OANO             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DAR5             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DMATCR5          | _          | _          | _          | _          | _          | _          | _         | _         |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| DMAC   | CHCR5            | TC         | [1:0]      |            | RLD1       | RLD        | 2[1:0]     | _         | IFT       |
|        |                  |            |            |            |            |            | HIE        |           |           |
|        |                  | DM         | [1:0]      | SM         | [1:0]      |            | RS         | [3:0]     | 1         |
|        |                  | _          | _          | тв         | TS[        | [1:0]      | IE         |           | DE        |
|        | CHFR5            | _          | _          | _          | HE         | _          | _          | _         | TE        |
|        | TEMSK5           |            |            |            | TEMK       | EY[7:0]    |            |           |           |
|        |                  | _          | _          | _          | _          | _          | _          |           | TEMASK    |
|        | RSAR5            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RDAR5            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RDMATCR5         |            |            | _          | _          |            | _          |           | _         |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ARCR5            |            |            | _          | _          |            | _          |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RARCR5           |            |            | _          | _          |            | _          |           | _         |
|        |                  |            |            |            |            |            |            |           |           |
|        | SAR6             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DAR6             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| DMAC   | DMATCR6          | _          |            |            | _          | _          | _          |           | _         |
|        |                  |            |            |            |            |            |            |           |           |
|        | CHCR6            | TC         | [1:0]      |            | RLD1       | RLD        | 2[1:0]     | _         | IFT       |
|        |                  |            |            | _  -  -    |            | _          | HIE        |           | _         |
|        |                  | DM         | [1:0]      | SM[1:0]    |            | RS[        |            | 3:0]<br>T | 1         |
|        |                  |            |            | ТВ         |            | [1:0]      | IE         |           | DE        |
|        | CHFR6            | _          | _          | _          | HE         | <u> </u>   |            | _         | TE        |
|        | TEMSK6           |            | 1          | 1          | TEMKE      | EY[7:0]    | ı          | ı         | 1         |
|        |                  | _          | _          | _          | _          | _          | _          | _         | TEMASK    |
|        | RSAR6            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | DDADO            |            |            |            |            |            |            |           |           |
|        | RDAR6            |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RDMATCR6         | _          | _          | _          | _          | _          | _          | _         |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ARCR6            | _          | _          | _          | _          | _          | _          | _         |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | RARCR6           |            | _          | _          | _          | _          | _          | _         |           |
|        | SAR7             |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2    | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|---------------|-----------|-----------|
| DMAC   | DAR7             |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | DMATCR7          | _          | _          | _          | _          | _          | _             | _         | _         |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | 011007           | TO:        | 1 01       |            | DI DI      | DI DI      | 254 03        |           |           |
|        | CHCR7            | 10         | [1:0]      | _          | RLD1       | RLD:       | 2[1:0]<br>HIE | _         | IFT       |
|        |                  | DM         | [1:0]      | SM         | [1:0]      |            | RS[           | 3.01      |           |
|        |                  |            |            | ТВ         |            | 1:0]       | IE            |           | DE        |
|        | CHFR7            | _          |            | _          | HE         | _          | _             |           | TE        |
|        | TEMSK7           |            |            | <u> </u>   | TEMKE      |            |               |           | 1         |
|        |                  | _          | _          | _          | _          |            | _             | _         | TEMASK    |
|        | RSAR7            |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | RDAR7            |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | RDMATCR7         |            |            |            |            |            |               |           |           |
|        | NDWATCH/         | _          |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        |                  |            |            |            |            |            |               |           |           |
|        | ARCR7            | _          | _          | _          | _          | _          | _             | _         | _         |
|        |                  |            |            |            |            |            |               |           |           |
|        | RARCR7           | _          | _          | _          | _          | _          | _             | _         | _         |
|        |                  |            |            |            |            |            |               |           |           |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| DMAC   | DMAOR            | _          | _          | CMS        | S[1:0]     | _          | _          | PR[       | 1:0]      |
|        |                  | _          | _          | _          | _          | _          | _          | _         | DME       |
|        | DMAFR            | _          | _          | _          | AE         | _          | _          | _         | NMIF      |
|        | DMARS0           |            |            | CH1 M      | IID[5:0]   |            |            | CH1 R     | ID[1:0]   |
|        |                  |            |            | CH0 M      | 1ID[5:0]   |            |            | CH0 R     | ID[1:0]   |
|        | DMARS1           |            |            | CH3 M      | 1ID[5:0]   |            |            | CH3 R     | ID[1:0]   |
|        |                  |            |            | CH2 M      | 1ID[5:0]   |            |            | CH2 R     | ID[1:0]   |
|        | DMARS2           |            |            | CH5 M      | 1ID[5:0]   |            |            | CH5 R     | ID[1:0]   |
|        |                  |            |            | CH4 M      | 1ID[5:0]   |            |            | CH4 R     | ID[1:0]   |
|        | DMARS3           |            |            | CH7 M      | 1ID[5:0]   |            |            | CH7 R     | ID[1:0]   |
|        |                  |            |            | CH6 M      | 1ID[5:0]   |            |            | CH6 R     | ID[1:0]   |
| A-DMAC | ADMAOR           |            |            |            |            |            |            |           | DME       |
|        | ADMAABR          |            | _          | _          | _          | _          |            | AA[2:0]   |           |
|        | ADMAIE0          |            |            |            |            |            |            | _         | _         |
|        | ADMAIE1          |            |            |            |            |            |            |           |           |
|        | ADMAIE2          |            |            |            |            |            |            |           |           |
|        | ADMAIE3          |            |            |            |            |            |            |           |           |
|        | ADMAIE4          |            |            |            |            |            |            |           |           |
|        | ADMAIE5          |            |            |            |            |            |            |           |           |
|        | ADMAIE7          |            |            |            |            |            |            |           |           |
|        | ADMAIE8          |            |            |            |            |            |            |           |           |
|        | ADMAIE9          | _          | _          | _          | _          | _          |            | _         | _         |
|        | ADMADV0          |            |            |            |            |            |            | _         | _         |
|        | ADMADV1          |            |            |            |            |            |            |           |           |
|        | ADMADV2          |            |            |            |            |            |            |           |           |
|        | ADMADV3          |            |            |            |            |            |            |           |           |
|        | ADMADV4          |            |            |            |            |            |            |           |           |
|        | ADMADV5          |            |            |            |            |            |            |           |           |
|        | ADMATE0          |            |            |            |            |            |            |           |           |
|        | ADMATE1          |            |            |            |            |            |            |           |           |
|        | ADMATE2          | _          | _          | _          | _          | _          |            | _         | _         |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| A-DMAC | ADMADE0          |            |            |            |            |            |            | _         | _         |
|        | ADMADE1          |            |            |            |            |            |            |           |           |
|        | ADMADE2          |            |            |            |            |            |            |           |           |
|        | ADMADE3          |            |            |            |            |            |            |           |           |
|        | ADMADE4          |            |            |            |            |            |            |           |           |
|        | ADMADE5          |            |            |            |            |            |            |           |           |
|        | ADMADE7          | _          | _          | _          | _          | _          |            | _         |           |
|        | ADMAMODE0        |            |            |            |            | _          | _          | _         | _         |
|        | ADMAMODE1        |            |            |            |            |            |            |           |           |
|        | ADMAMODE2        |            |            |            |            |            |            |           |           |
|        | ADMATCR0         | _          | _          | _          | _          | _          |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMARTCR0        | _          | _          | _          | _          | _          | _          |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMATCR1         | _          | _          | _          | _          | _          | _          |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMARTCR1        | _          | _          | _          | _          | _          | _          |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMATCR56        | _          |            | _          |            |            | _          |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMATCR57        | _          | _          | _          | _          | _          | _          |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMATCR58        | _          |            | _          | _          |            | _          |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMATCR59        | _          |            | _          |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMATCR60        |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMATCR61        |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| A-DMAC | ADMATCR62        | _          | _          | _          | _          | _          | _          |           |           |
|        | ADMATCR63        | _          | _          | _          | _          | _          | _          |           |           |
|        | ADMATCR64        | _          |            |            |            |            |            |           |           |
|        | ADMATCR65        | _          | _          | _          | _          | _          | _          |           |           |
|        | ADMATCR66        | _          | _          | _          | _          | _          | _          |           |           |
|        | ADMATCR67        | _          |            |            |            | _          |            |           |           |
|        | ADMATCR68        | _          | _          | _          | _          | _          | _          |           |           |
|        | ADMATCR69        | _          | _          | _          | _          | _          | _          |           |           |
|        | ADMATCR70        | _          | _          | _          | _          | _          | _          |           |           |
|        | ADMATCR71        | _          | _          | _          | _          | _          | _          |           |           |
|        | ADMAAR0          | _          |            |            |            |            |            |           |           |
|        | ADMARAR0         | _          |            |            |            |            |            |           |           |
|        | ADMAAR1          | _          |            |            |            |            |            |           |           |
|        | ADMARAR1         | _          |            |            |            |            |            |           |           |
|        | ADMAAR56         |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6  | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0  |
|--------|------------------|------------|-------------|------------|------------|------------|------------|-----------|------------|
|        |                  | 01/20/10/1 | 00/22/1-1/0 | 20/21/10/0 | 20/20/12/4 | 2171071170 | 20/10/10/2 | 20/11/0/1 | 2-1/10/0/0 |
| A-DMAC | ADMAAR57         |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR58         |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR59         |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR60         | _          |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR61         |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR62         | _          |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR63         |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR64         |            |             |            |            |            |            |           |            |
|        | 7 DW/VIIIO4      |            |             |            |            |            |            |           |            |
|        | ADMAAR65         | _          |             |            |            |            |            |           |            |
|        | ADMAAR65         |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR66         |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR67         |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR68         |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR69         | _          |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR70         | _          |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        | ADMAAR71         | _          |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |
|        |                  |            |             |            |            |            |            |           |            |

RENESAS

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| A-DMAC | ADMABUF2         |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMABUF3         |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMABUF4         |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMABUF5         |            |            |            |            |            |            |           |           |
|        | ADMADOI 3        |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMABUF6         |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMABUF7         |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMARVPR0        |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMARVPR1        |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           |           |
|        | ADMARVPR2        |            |            |            |            |            |            |           |           |
|        |                  |            |            |            |            |            |            |           | <u> </u>  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|---------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| A-DMAC  | ADMARVPR3        |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         | ADMARVPR4        |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         | ADMARVPR5        |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         | ADMATVPR0        |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         | ADMATVPR1        |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         | ADMATVPR2        |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         | ADMATVPR3        |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         | ADMATVPR4        |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         | ADMATVPR5        |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
| ATU-III | ATUENR           | _          | _          | _          | _          | _          | _          | TJE       | THE       |
|         |                  | TGE        | TFE        | TEE        | TDE        | TCE        | TBE        | TAE       | PSCE      |
|         | CBCNT            | _          |            | CB4E       | G[1:0]     |            | CB5SEL     | CB5E      | G[1:0]    |
|         | NCMR             | NCCSEL     | _          | _          | _          | NCMJ       | NCMF       | NCMC      | NCMA      |
|         | PSCR0            | _          |            | _          | _          |            | _          | PSC       | 0[9:8]    |
|         |                  |            |            |            | PSC        | [7:0]      |            |           |           |
|         | PSCR1            | _          | _          | _          |            | _          | _          | PSC.      | 1[9:8]    |
|         |                  |            |            |            | PSC1       | [7:0]      |            |           |           |
|         | PSCR2            |            |            |            |            |            |            | PSC       | 2[9:8]    |
|         |                  |            |            |            |            |            |            |           |           |
|         | PSCR3            | _          | _          | _          |            | _          | _          | PSC       | 3[9:8]    |
|         |                  |            |            |            | PSC3       | 3[7:0]     |            |           |           |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3     | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|--------------|--------------|------------|------------|----------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | TCRA             | EVOSEL2A     | EVOSEL2B     |            | EVOSEL1    | ı              |            | CKSELA    |           |  |  |  |  |
|         | TIOR1A           | _            | _            | _          | _          | IO             | <b>A</b> 5 | IC        | )A4       |  |  |  |  |
|         |                  | 10           | A3           | IO         | A2         | Ю              | A1         | IOA0      |           |  |  |  |  |
|         | TIOR2A           | _            | _            | NCKA5      | NCKA4      | NCKA3          | NCKA2      | NCKA1     | NCKA0     |  |  |  |  |
|         |                  |              | _            | NCEA5      | NCEA4      | NCEA3          | NCEA2      | NCEA1     | NCEA0     |  |  |  |  |
|         | TSRA             | OVFA         | _            | ICFA5      | ICFA4      | ICFA3          | ICFA2      | ICFA1     | ICFA0     |  |  |  |  |
|         | TIERA            | OVEA         | _            | ICEA5      | ICEA4      | ICEA3          | ICEA2      | ICEA1     | ICEA0     |  |  |  |  |
|         | NCNTA0           |              | NCCNTA0[7:0] |            |            |                |            |           |           |  |  |  |  |
|         | NCRA0            |              |              |            | NCTA       | .0[7:0]        |            |           |           |  |  |  |  |
|         | NCNTA1           |              |              |            | NCCNT      | A1[7:0]        |            |           |           |  |  |  |  |
|         | NCRA1            |              |              |            | NCTA       | 1[7:0]         |            |           |           |  |  |  |  |
|         | NCNTA2           |              |              |            | NCCNT      | A2[7:0]        |            |           |           |  |  |  |  |
|         | NCRA2            |              | NCTA2[7:0]   |            |            |                |            |           |           |  |  |  |  |
|         | NCNTA3           |              | NCCNTA3[7:0] |            |            |                |            |           |           |  |  |  |  |
|         | NCRA3            | NCTA3[7:0]   |              |            |            |                |            |           |           |  |  |  |  |
|         | NCNTA4           | NCCNTA4[7:0] |              |            |            |                |            |           |           |  |  |  |  |
|         | NCRA4            |              |              |            | NCTA       | 4[7:0]         |            |           |           |  |  |  |  |
|         | NCNTA5           |              |              |            | NCCNT      | A5[7:0]        |            |           |           |  |  |  |  |
|         | NCRA5            |              |              |            | NCTA       | 5[7:0]         |            |           |           |  |  |  |  |
|         | TCNTA            |              |              |            | CNTA       | [31:24]        |            |           |           |  |  |  |  |
|         |                  |              |              |            | CNTA       | [23:16]        |            |           |           |  |  |  |  |
|         |                  |              |              |            | CNTA       | [15:8]         |            |           |           |  |  |  |  |
|         |                  |              |              |            | CNT        | <b>A</b> [7:0] |            |           |           |  |  |  |  |
|         | ICRA0            |              |              |            | ICA0[      | 31:24]         |            |           |           |  |  |  |  |
|         |                  |              |              |            | ICA0[      | 23:16]         |            |           |           |  |  |  |  |
|         |                  | ICA0[15:8]   |              |            |            |                |            |           |           |  |  |  |  |
|         |                  |              |              |            | ICAC       | 0[7:0]         |            |           |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6  | 29/21/13/5 | 28/20/12/4            | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0   |  |  |  |  |
|---------|------------------|------------|-------------|------------|-----------------------|------------|------------|-----------|-------------|--|--|--|--|
| ATU-III | ICRA1            |            | <u>I</u>    | <u>I</u>   | ICA1[                 | 31:24]     | ı          | <u>I</u>  |             |  |  |  |  |
|         |                  |            |             |            | ICA1[                 | 23:16]     |            |           |             |  |  |  |  |
|         |                  |            |             |            | ICA1                  | 15:8]      |            |           |             |  |  |  |  |
|         |                  |            |             |            | ICA1                  | [7:0]      |            |           |             |  |  |  |  |
|         | ICRA2            |            |             |            | ICA2[                 | 31:24]     |            |           |             |  |  |  |  |
|         |                  |            |             |            | ICA2[                 | 23:16]     |            |           |             |  |  |  |  |
|         |                  |            |             |            | ICA2                  | 15:8]      |            |           |             |  |  |  |  |
|         |                  |            |             |            | ICA2                  | [7:0]      |            |           |             |  |  |  |  |
|         | ICRA3            |            | ICA3[31:24] |            |                       |            |            |           |             |  |  |  |  |
|         |                  |            | ICA3[23:16] |            |                       |            |            |           |             |  |  |  |  |
|         |                  |            |             |            | ICA3                  | 15:8]      |            |           |             |  |  |  |  |
|         |                  |            |             |            | ICA3                  | [7:0]      |            |           |             |  |  |  |  |
|         | ICRA4            |            | ICA4[31:24] |            |                       |            |            |           |             |  |  |  |  |
|         |                  |            | ICA4[23:16] |            |                       |            |            |           |             |  |  |  |  |
|         |                  |            | ICA4[15:8]  |            |                       |            |            |           |             |  |  |  |  |
|         |                  |            | ICA4[7:0]   |            |                       |            |            |           |             |  |  |  |  |
|         | ICRA5            |            |             |            | ICA5[                 |            |            |           |             |  |  |  |  |
|         |                  |            |             |            | ICA5[                 |            |            |           |             |  |  |  |  |
|         |                  |            |             |            | ICA5                  |            |            |           |             |  |  |  |  |
|         | TODD             |            | <u> </u>    | <u> </u>   | ICA5                  | [7:0]<br>  | 1          | 0145      |             |  |  |  |  |
|         | TCRB             | -          | OTONITOS    |            | -                     | _          |            | CKS       | ELB<br>Long |  |  |  |  |
|         | TIORB            | LDSEL      | CTCNTB5     | EVCNTB     | LDEN                  | CCS        | - CMED4    |           | IOB6        |  |  |  |  |
|         | TSRB             |            |             |            | <u> </u>              | CMFB6      | CMFB1      | ICFB0     | CMFB0       |  |  |  |  |
|         | TIERB TCNTB0     |            |             | in         | EG<br>CN <sup>-</sup> | CMEB6      | CMEB1      | ICEB0     | CMEB0       |  |  |  |  |
|         | TONTBU           |            |             |            |                       |            |            |           |             |  |  |  |  |
|         |                  | CNTB0      |             |            |                       |            |            |           |             |  |  |  |  |
|         |                  |            | CNTBO       |            |                       |            |            |           |             |  |  |  |  |
|         |                  |            |             |            | CN                    | I DU       |            |           |             |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | ICRB0            |            |            | ·          | ICI        | 30         |            | Į.        | U.        |  |  |  |  |
|         |                  |            |            |            | ICI        | 30         |            |           |           |  |  |  |  |
|         |                  |            |            |            | ICI        | 30         |            |           |           |  |  |  |  |
|         |                  |            |            |            | ICI        | 30         |            |           |           |  |  |  |  |
|         | OCRB0            |            |            |            | oc         | B0         |            |           |           |  |  |  |  |
|         |                  |            |            |            | OC         | B0         |            |           |           |  |  |  |  |
|         |                  |            |            |            | OC         | B0         |            |           |           |  |  |  |  |
|         |                  |            | OCB0       |            |            |            |            |           |           |  |  |  |  |
|         | TCNTB1           |            | CNTB1      |            |            |            |            |           |           |  |  |  |  |
|         | OCRB1            |            |            |            | OC         |            |            |           |           |  |  |  |  |
|         | ICRB1            |            |            |            | ICI        |            |            |           |           |  |  |  |  |
|         |                  |            | ICB1       |            |            |            |            |           |           |  |  |  |  |
|         |                  |            | ICB1       |            |            |            |            |           |           |  |  |  |  |
|         |                  |            | ICB1       |            |            |            |            |           |           |  |  |  |  |
|         | ICRB2            | ICB2       |            |            |            |            |            |           |           |  |  |  |  |
|         |                  | ICB2       |            |            |            |            |            |           |           |  |  |  |  |
|         |                  |            | ICB2       |            |            |            |            |           |           |  |  |  |  |
|         | 1.00             |            | 1          |            | ICI        | 32         | 1          | 1         | 1         |  |  |  |  |
|         | LDB              | _          | _          |            | LD\        | <u> </u>   | _          |           |           |  |  |  |  |
|         |                  |            |            |            | LD\        |            |            |           |           |  |  |  |  |
|         |                  |            |            |            | LD\        |            |            |           |           |  |  |  |  |
|         | RLDB             |            |            |            | RLD        |            |            |           |           |  |  |  |  |
|         | NLDD             |            |            |            | RLD        |            |            |           |           |  |  |  |  |
|         |                  |            |            |            | RLD        |            |            |           |           |  |  |  |  |
|         |                  |            |            |            |            |            |            | _         | _         |  |  |  |  |
|         | PIMR             | PIM        |            |            |            |            |            |           |           |  |  |  |  |
|         |                  |            | <u> </u>   |            | l<br>Pl    | PIM        |            |           |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7  | 30/22/14/6     | 29/21/13/5 | 28/20/12/4 | 27/19/11/3      | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |
|---------|------------------|-------------|----------------|------------|------------|-----------------|------------|-----------|-----------|--|--|--|
| ATU-III | TCNTB2           |             | l              | <u>I</u>   | CNT        | TB2             | <u>I</u>   | l         |           |  |  |  |
|         |                  |             |                |            | CNT        | ГВ2             |            |           |           |  |  |  |
|         |                  |             |                |            | CNT        | ГВ2             |            |           |           |  |  |  |
|         |                  | _           |                |            |            |                 |            |           |           |  |  |  |
|         | TCNTB6           |             |                |            | CNT        | ГВ6             |            |           |           |  |  |  |
|         |                  |             |                |            | CNT        | TB6             | T          | T         | _         |  |  |  |
|         |                  |             | CN             | TB6        | 1          | _               | _          | _         | _         |  |  |  |
|         |                  | _           |                | _          | _          |                 | _          | _         |           |  |  |  |
|         | OCRB6            |             | OCB6           |            |            |                 |            |           |           |  |  |  |
|         |                  |             | OCB6           |            |            |                 |            |           |           |  |  |  |
|         |                  |             | OCB6 — — —     |            |            |                 |            |           |           |  |  |  |
|         |                  | _           |                |            |            | _               |            |           | _         |  |  |  |
|         | OCRB7            |             | OCB7           |            |            |                 |            |           |           |  |  |  |
|         |                  |             | OCB7 — — — — — |            |            |                 |            |           |           |  |  |  |
|         |                  |             | T              | ,в,        | 1          |                 |            |           | _         |  |  |  |
|         | TCNTB3           |             |                |            |            |                 |            |           |           |  |  |  |
|         | TONTES           |             | CNTB3          |            |            |                 |            |           |           |  |  |  |
|         |                  |             | CN             | TB3        |            |                 |            |           |           |  |  |  |
|         |                  |             | _              | _          | _          | _               |            | _         | _         |  |  |  |
|         | TCNTB4           |             |                |            | CNT        | <u>I</u><br>ГВ4 |            | l         |           |  |  |  |
|         |                  |             |                |            | CNT        | ГВ4             |            |           |           |  |  |  |
|         |                  | CNTB4 — — — |                |            |            |                 |            |           |           |  |  |  |
|         |                  | _           | _              | _          | _          | _               | _          | _         | _         |  |  |  |
|         | TCNTB5           |             | •              |            | CNT        | ГВ5             |            | •         |           |  |  |  |
|         |                  |             | CNTB5          |            |            |                 |            |           |           |  |  |  |
|         |                  |             | CN             | TB5        |            | _               | _          | _         | _         |  |  |  |
|         |                  |             |                |            |            |                 |            |           |           |  |  |  |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6                   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|--------------|------------------------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | TCCLRB           |              | I                            | I          | CCI        | _RB        | I          | I         | II.       |  |  |  |  |
|         |                  |              |                              |            | CCI        | _RB        |            |           |           |  |  |  |  |
|         |                  |              | CC                           | LRB        |            | _          | _          | _         | _         |  |  |  |  |
|         |                  |              | _                            | _          | _          |            | _          |           |           |  |  |  |  |
|         | TSTRC            | _            | _                            | _          | STRC4      | STRC3      | STRC2      | STRC1     | STRC0     |  |  |  |  |
|         | NCCRC0           | _            | _                            | _          | _          | NCEC03     | NCEC02     | NCEC01    | NCEC00    |  |  |  |  |
|         | NCCRC1           | _            | _                            | _          | _          | NCEC13     | NCEC12     | NCEC11    | NCEC10    |  |  |  |  |
|         | NCCRC2           | _            | _                            | _          | _          | NCEC23     | NCEC22     | NCEC21    | NCEC20    |  |  |  |  |
|         | NCCRC3           | _            | _                            | _          | _          | NCEC33     | NCEC32     | NCEC31    | NCEC30    |  |  |  |  |
|         | NCCRC4           | _            | _                            | _          | _          | NCEC43     | NCEC42     | NCEC41    | NCEC40    |  |  |  |  |
|         | NCNTC00          |              | NCNTC00[7:0]                 |            |            |            |            |           |           |  |  |  |  |
|         | NCNTC01          |              | NCNTC01[7:0]<br>NCNTC02[7:0] |            |            |            |            |           |           |  |  |  |  |
|         | NCNTC02          |              |                              |            |            |            |            |           |           |  |  |  |  |
|         | NCNTC03          |              | NCNTC03[7:0]                 |            |            |            |            |           |           |  |  |  |  |
|         | NCRC00           |              | NCRC00[7:0]                  |            |            |            |            |           |           |  |  |  |  |
|         | NCRC01           |              | NCRC01[7:0]                  |            |            |            |            |           |           |  |  |  |  |
|         | NCRC02           | NCRC02[7:0]  |                              |            |            |            |            |           |           |  |  |  |  |
|         | NCRC03           | NCRC03[7:0]  |                              |            |            |            |            |           |           |  |  |  |  |
|         | NCNTC10          | NCNTC10[7:0] |                              |            |            |            |            |           |           |  |  |  |  |
|         | NCNTC11          |              |                              |            | NCNTO      | 11[7:0]    |            |           |           |  |  |  |  |
|         | NCNTC12          |              |                              |            | NCNTO      | 12[7:0]    |            |           |           |  |  |  |  |
|         | NCNTC13          |              |                              |            | NCNTO      | 13[7:0]    |            |           |           |  |  |  |  |
|         | NCRC10           |              |                              |            | NCRC       | 10[7:0]    |            |           |           |  |  |  |  |
|         | NCRC11           |              |                              |            | NCRC       | 11[7:0]    |            |           |           |  |  |  |  |
|         | NCRC12           |              |                              |            | NCRC       | 12[7:0]    |            |           |           |  |  |  |  |
|         | NCRC13           |              |                              |            | NCRC       | 13[7:0]    |            |           |           |  |  |  |  |
|         | NCNTC20          |              |                              |            | NCNTC      | 20[7:0]    |            |           |           |  |  |  |  |
|         | NCNTC21          |              |                              |            | NCNTC      | 21[7:0]    |            |           |           |  |  |  |  |
|         | NCNTC22          |              |                              |            | NCNTC      | 22[7:0]    |            |           |           |  |  |  |  |
|         | NCNTC23          |              |                              |            | NCNTC      | 23[7:0]    |            |           |           |  |  |  |  |
|         | NCRC20           |              |                              |            | NCRC       | 20[7:0]    |            |           |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |  |  |  |  |
|---------|------------------|------------|--------------|------------|------------|------------|------------|------------|-----------|--|--|--|--|
| ATU-III | NCRC21           |            |              | <u>I</u>   | NCRC       | 21[7:0]    | ı          | <u>I</u>   | l         |  |  |  |  |
|         | NCRC22           |            |              |            | NCRC       | 22[7:0]    |            |            |           |  |  |  |  |
|         | NCRC23           |            |              |            | NCRC       | 23[7:0]    |            |            |           |  |  |  |  |
|         | NCNTC30          |            |              |            | NCNTC      | 30[7:0]    |            |            |           |  |  |  |  |
|         | NCNTC31          |            |              |            | NCNTO      | 31[7:0]    |            |            |           |  |  |  |  |
|         | NCNTC32          |            |              |            | NCNTO      | 32[7:0]    |            |            |           |  |  |  |  |
|         | NCNTC33          |            |              |            | NCNTC      | 33[7:0]    |            |            |           |  |  |  |  |
|         | NCRC30           |            | NCRC30[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | NCRC31           |            | NCRC31[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | NCRC32           |            | NCRC32[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | NCRC33           |            |              |            | NCRC       | 33[7:0]    |            |            |           |  |  |  |  |
|         | NCNTC40          |            |              |            | NCNTO      | 240[7:0]   |            |            |           |  |  |  |  |
|         | NCNTC41          |            | NCNTC41[7:0] |            |            |            |            |            |           |  |  |  |  |
|         | NCNTC42          |            | NCNTC42[7:0] |            |            |            |            |            |           |  |  |  |  |
|         | NCNTC43          |            | NCNTC43[7:0] |            |            |            |            |            |           |  |  |  |  |
|         | NCRC40           |            | NCRC40[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | NCRC41           |            | NCRC41[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | NCRC42           |            |              |            | NCRC       | 42[7:0]    |            |            |           |  |  |  |  |
|         | NCRC43           |            |              |            | NCRC       | 43[7:0]    |            |            |           |  |  |  |  |
|         | TCRC0            | FCMC03     | FCMC02       | FCMC01     | FCMC00     | PWM00      | C          | KSELC0[2:  | 0]        |  |  |  |  |
|         | TIERC0           | _          | _            | _          | OVEC0      | IMEC03     | IMEC02     | IMEC01     | IMEC00    |  |  |  |  |
|         | TIORC0           | _          |              | IOC03[2:0] |            | _          |            | IOC02[2:0] |           |  |  |  |  |
|         |                  | _          |              | IOC01[2:0] |            | _          |            | IOC00[2:0] |           |  |  |  |  |
|         | TSRC0            | _          | _            | _          | OVFC0      | IMFC03     | IMFC02     | IMFC01     | IMFC00    |  |  |  |  |
|         | GRC00            |            |              |            |            |            |            |            |           |  |  |  |  |
|         |                  |            |              |            |            |            |            |            |           |  |  |  |  |
|         |                  |            |              |            |            |            |            |            |           |  |  |  |  |
|         |                  | _          | _            | _          | _          | _          | _          | _          |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|---------|------------------|------------|------------|------------|------------|------------|------------|------------|-----------|
| ATU-III | GRC01            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|         | GRC02            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|         | GRC03            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            | _          | _         |
|         | TCNTC0           |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          |            |            | _         |
|         | TCRC1            | FCMC13     | FCMC12     | FCMC11     | FCMC10     | PWM10      | C          | KSELC1[2:0 | 0]        |
|         | TIERC1           | _          | _          | _          | OVEC1      | IMEC13     | IMEC12     | IMEC11     | IMEC10    |
|         | TIORC1           | _          |            | IOC13[2:0] |            |            |            | IOC12[2:0] |           |
|         |                  | _          |            | IOC11[2:0] |            |            |            | IOC10[2:0] |           |
|         | TSRC1            | _          | _          | _          | OVFC1      | IMFC13     | IMFC12     | IMFC11     | IMFC10    |
|         | GRC10            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          |           |
|         | GRC11            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          |            |            | _          | _          | _          | _          | _         |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|---------|------------------|------------|------------|------------|------------|------------|------------|------------|-----------|
| ATU-III | GRC12            |            |            |            |            |            |            |            |           |
| A10-III | GNOTZ            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         | GRC13            |            |            |            |            |            |            |            |           |
|         | GHC13            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            | <u> </u>   | <u> </u>   | _          | _          | _          | <u> </u>   | _         |
|         | TCNTC1           |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|         | TCRC2            | FCMC23     | FCMC22     | FCMC21     | FCMC20     | PWM20      | С          | 0]         |           |
|         | TIERC2           |            |            |            |            | IMEC23     | IMEC22     | IMEC21     | IMEC20    |
|         | TIORC2           | _          | IOC23[2:0] |            |            | _          |            | IOC22[2:0] |           |
|         |                  | _          |            | IOC21[2:0] |            |            | IOC20[2:0] |            |           |
|         | TSRC2            | _          | _          | _          | OVFC2      | IMFC23     | IMFC22     | IMFC21     | IMFC20    |
|         | GRC20            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          |            | _          | _         |
|         | GRC21            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            | _         |
|         | GRC22            |            |            |            |            |            |            |            | 1         |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|---------|------------------|------------|------------|------------|------------|------------|------------|------------|-----------|
| ATU-III | GRC23            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|         | TCNTC2           |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|         | TCRC3            | FCMC33     | FCMC32     | FCMC31     | FCMC30     | PWM30      | C          | KSELC3[2:  | 0]        |
|         | TIERC3           | _          | _          | _          | OVEC3      | IMEC33     | IMEC32     | IMEC31     | IMEC30    |
|         | TIORC3           | _          |            | IOC33[2:0] |            | _          |            | IOC32[2:0] |           |
|         |                  | _          |            | IOC31[2:0] |            | _          |            | IOC30[2:0] |           |
|         | TSRC3            | _          |            |            | OVFC3      | IMFC33     | IMFC32     | IMFC31     | IMFC30    |
|         | GRC30            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|         | GRC31            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|         | GRC32            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          |            |            | _          |            | _          | _         |
|         | GRC33            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          |            |            |            |            |            | _          | _         |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|---------|------------------|------------|------------|------------|------------|------------|------------|------------|-----------|
| ATU-III | TCNTC3           |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|         | TCRC4            | FCMC43     | FCMC42     | FCMC41     | FCMC40     | PWM40      | C          | KSELC4[2:  | 0]        |
|         | TIERC4           | _          | _          | _          | OVEC4      | IMEC43     | IMEC42     | IMEC41     | IMEC40    |
|         | TIORC4           | _          |            | IOC43[2:0] |            | _          |            | IOC42[2:0] |           |
|         |                  | _          |            | IOC41[2:0] |            | _          |            | IOC40[2:0] |           |
|         | TSRC4            | _          |            |            | OVFC4      | IMFC43     | IMFC42     | IMFC41     | IMFC40    |
|         | GRC40            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          |            |            | _          |            | _         |
|         | GRC41            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          | _          | _          | _          | _          | _          | _          | _         |
|         | GRC42            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | _          |            | _          | _          |            | _          | _          | _         |
|         | GRC43            |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  | —          | _          | _          | _          |            | _          |            | —         |
|         | TCNTC4           |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         |                  |            |            |            |            |            |            |            |           |
|         | TSTRD            | _          | _          | _          | _          | STRD3      | STRD2      | STRD1      | STRD0     |

RENESAS

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5  | 28/20/12/4 | 27/19/11/3    | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|---------|------------------|------------|------------|-------------|------------|---------------|------------|------------|-----------|
| ATU-III | TCNT1D0          |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  | _          | _          | _           | _          | _             | _          | _          | _         |
|         | TCNT2D0          |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            | _         |
|         | OSBRD0           |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         | TCRD0            |            | OBRED0     | C2CED0      | C1CED0     |               | C          | ∪1         |           |
|         | TONDO            |            |            | KSEL1D0[2:  |            |               |            | KSEL2D0[2: |           |
|         | TOCRD0           |            |            | NOLL IDO[2. |            |               | _          |            | TONEAD0   |
|         | CMPOD0           | CMDDDOO    | CMDBD00    | CMDDDO1     | CMDBD00    | CMPAD03       | CMDADOO    |            |           |
|         |                  | CIVIPBDU3  | CIVIPBDU2  | CIVIPBDUT   | CMPBD00    | CMPADUS       | CMPAD02    | CIVIPADUT  | CIVIPADOO |
|         | TCNT1D1          |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         | TCNT2D1          |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            | _          |             | _          | _             |            |            | _         |
|         | OSBRD1           |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  |            |            |             |            |               |            |            |           |
|         |                  | _          | _          | _           | _          | _             | _          |            | _         |
|         | TCRD1            |            | OBRED1     | C2CED1      | C1CED1     | CKSEL2D1[2:0] |            |            |           |
|         |                  |            | С          | KSEL1D1[2:  | [0]        |               |            | CSELD1[2:  | 0]        |
|         | TOCRD1           |            |            |             |            |               |            | TONEBD1    | TONEAD1   |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6                  | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2    | 25/17/9/1 | 24/16/8/0 |  |
|---------|------------------|------------|-----------------------------|------------|------------|------------|---------------|-----------|-----------|--|
| ATU-III | CMPOD1           | CMPBD13    | CMPBD12                     | CMPBD11    | CMPBD10    | CMPAD13    | CMPAD12       | CMPAD11   | CMPAD10   |  |
|         | TCNT1D2          |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  | _          | _                           | _          | _          | _          | _             | _         | _         |  |
|         | TCNT2D2          |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  | _          | _                           | _          | _          | _          | _             | _         | _         |  |
|         | OSBRD2           |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             | _          | _          |            | _             |           | _         |  |
|         | TCRD2            | _          | OBRED2                      | C2CED2     | C1CED2     | _          | CKSEL2D2[2:0] |           |           |  |
|         |                  |            | С                           | KSEL1D2[2: | 0]         |            | DCSELD2[2:0]  |           |           |  |
|         | TOCRD2           |            |                             | _          | _          |            | _             | TONEBD2   | TONEAD2   |  |
|         | TCNT1D3          |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  | _          |                             | _          |            |            | _             | _         |           |  |
|         | TCNT2D3          |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  | _          | _                           | _          | _          | _          | _             | _         | _         |  |
|         | OSBRD3           |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  |            |                             |            |            |            |               |           |           |  |
|         |                  | _          | _                           | _          | _          | _          | _             | _         | _         |  |
|         | TCRD3            |            | OBRED3                      | C2CED3     | C1CED3     | _          | CKSEL2D3[2:0] |           |           |  |
|         |                  |            | CKSEL1D3[2:0] — DCSELD3[2:0 |            |            |            |               |           | 0]        |  |
|         | TOCRD3           | _          | _                           | _          | _          | _          | _             | TONEBD3   | TONEAD3   |  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1   | 24/16/8/0 |
|---------|------------------|------------|------------|------------|------------|------------|------------|-------------|-----------|
| ATU-III | TIOR1D0          | OSSD       | 03[1:0]    | OSSD       | 02[1:0]    | OSSD       | 01[1:0]    | OSSD        | 00[1:0]   |
|         |                  | IOAD       | 03[1:0]    | IOAD       | 02[1:0]    | IOAD       | 01[1:0]    | IOAD        | 00[1:0]   |
|         | TIOR2D0          | _          |            | IOBD03[2:0 | ]          | _          |            | IOBD02[2:0] | ]         |
|         |                  | _          |            | IOBD01[2:0 | ]          | _          |            | IOBD00[2:0] | ]         |
|         | DSTRD0           | _          |            | _          | _          | DSTD03     | DSTD02     | DSTD01      | DSTD00    |
|         | DSRD0            | _          | _          | _          | _          | DSFD03     | DSFD02     | DSFD01      | DSFD00    |
|         | DCRD0            | _          | TF         | RGSELD03[2 | 2:0]       | _          | TF         | RGSELD02[2  | 2:0]      |
|         |                  | _          | TF         | RGSELD01[2 | 2:0]       | _          | TF         | RGSELD00[2  | 2:0]      |
|         | TSRD0            | _          | _          | OVF2D0     | OVF1D0     | UDFD03     | UDFD02     | UDFD01      | UDFD00    |
|         |                  | CMFAD03    | CMFAD02    | CMFAD01    | CMFAD00    | CMFBD03    | CMFBD02    | CMFBD01     | CMFBD00   |
|         | TIERD0           | _          | _          | OVE2D0     | OVE1D0     | UDED03     | UDED02     | UDED01      | UDED00    |
|         |                  | CMEAD03    | CMEAD02    | CMEAD01    | CMEAD00    | CMEBD03    | CMEBD02    | CMEBD01     | CMEBD00   |
|         | OCRD00           |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  |            | _          | _          | _          | _          | _          | _           | _         |
|         | OCRD01           |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  |            | _          | _          | _          | _          | _          | _           | _         |
|         | OCRD02           |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  | _          | _          | _          | _          | _          | _          | _           | _         |
|         | OCRD03           |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  | _          | _          |            | _          | _          | _          | _           |           |
|         | GRD00            |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  |            |            |            |            |            |            |             |           |
|         |                  | _          | _          | _          | _          | _          | _          | _           | _         |

|         | Register |            |            |            |            |             |            |             |           |
|---------|----------|------------|------------|------------|------------|-------------|------------|-------------|-----------|
| Module  | Name     | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3  | 26/18/10/2 | 25/17/9/1   | 24/16/8/0 |
| ATU-III | GRD01    |            |            |            |            |             |            |             |           |
|         |          |            |            |            |            |             |            |             |           |
|         |          |            | _          | _          | _          | _           | _          | _           | _         |
|         | GRD02    |            |            |            |            |             |            |             |           |
|         |          |            |            |            |            |             |            |             |           |
|         |          |            | _          | _          | _          | _           | _          | _           | _         |
|         | GRD03    |            |            |            |            |             |            |             |           |
|         |          |            |            |            |            |             |            |             |           |
|         |          |            |            |            |            |             |            |             |           |
|         | DCNTD00  |            |            |            |            |             |            |             |           |
|         |          |            |            |            |            |             |            |             |           |
|         |          |            |            |            |            |             |            |             |           |
|         | DONTDO   |            | _          | _          | _          | _           | _          | _           | _         |
|         | DCNTD01  |            |            |            |            |             |            |             |           |
|         |          |            |            |            |            |             |            |             |           |
|         |          |            | _          | _          | _          | _           | _          | _           | _         |
|         | DCNTD02  |            |            |            |            |             |            |             |           |
|         |          |            |            |            |            |             |            |             |           |
|         |          | _          | _          | _          | _          | _           | _          | _           | _         |
|         | DCNTD03  |            |            |            |            |             |            |             |           |
|         |          |            |            |            |            |             |            |             |           |
|         |          | _          | _          | _          | _          | _           | _          | _           | _         |
|         | TIOR1D1  | OSSD       | 13[1:0]    | OSSD       | 12[1:0]    | OSSD11[1:0] |            | OSSD10[1:0] |           |
|         |          | IOAD       | 13[1:0]    | IOAD       | 12[1:0]    | IOAD        | 11[1:0]    | IOAD10[1:0] |           |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5  | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|---------|------------------|------------|------------|-------------|------------|------------|------------|------------|-----------|
| ATU-III | TIOR2D1          | _          |            | IOBD13[2:0] |            | _          |            | IOBD12[2:0 |           |
|         |                  |            |            | IOBD11[2:0] |            | _          |            | IOBD10[2:0 | 1         |
|         | DSTRD1           | _          | _          |             |            | DSTD13     | DSTD12     | DSTD11     | DSTD10    |
|         | DSRD1            | _          |            |             |            | DSFD13     | DSFD12     | DSFD11     | DSFD10    |
|         | DCRD1            | _          | TF         | RGSELD13[2  | 2:0]       | _          | TF         | RGSELD12[2 | 2:0]      |
|         |                  | _          | TF         | RGSELD11[2  | 2:0]       | _          | TF         | RGSELD10[2 | 2:0]      |
|         | TSRD1            |            | _          | OVF2D1      | OVF1D1     | UDFD13     | UDFD12     | UDFD11     | UDFD10    |
|         |                  | CMFAD13    | CMFAD12    | CMFAD11     | CMFAD10    | CMFBD13    | CMFBD12    | CMFBD11    | CMFBD10   |
|         | TIERD1           | _          | _          | OVE2D1      | OVE1D1     | UDED13     | UDED12     | UDED11     | UDED10    |
|         |                  | CMEAD13    | CMEAD12    | CMEAD11     | CMEAD10    | CMEBD13    | CMEBD12    | CMEBD11    | CMEBD10   |
|         | OCRD10           |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  | _          | _          |             | _          |            |            | _          |           |
|         | OCRD11           |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  | _          | _          |             | _          | _          | _          | _          | _         |
|         | OCRD12           |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  | _          | _          | _           | _          | _          | _          | _          | _         |
|         | OCRD13           |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            | _          |             |            | _          |            |            |           |
|         | GRD10            |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |

| Register<br>Name | 31/23/15/7                                     | 30/22/14/6   | 29/21/13/5 | 28/20/12/4                                 | 27/19/11/3  | 26/18/10/2   | 25/17/9/1  | 24/16/8/0  |
|------------------|--|--------------|------------|--|-------------|--|--|--|
| GRD11            |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  | _            | _          | _  | _           |  | _  | _  |
| GRD12            |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  | _            | _          | _  | _           |  | _  | _  |
| GRD13            |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  | _            | _          | _  | _           |  | _  | _  |
| DCNTD10          |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  | _            | _          | _  | _           | _  | _  | _  |
| DCNTD11          |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  | _            | _          | _  | _           |  | _  | _  |
| DCNTD12          |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
| DCNTD13          |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  |              |            |  |             |  |  |  |
|                  |  |              | _          | _  |             |  | _  |  |
| TIOR1D2          | OSSD   | 1<br>23[1:0] | OSSD       | 1<br>22[1:0]                               | OSSD21[1:0] |  | OSSD20[1:0]  |  |
|                  |  |              |            |  |             |  | IOAD20[1:0]  |  |
|                  | GRD11  GRD12  GRD13  DCNTD10  DCNTD11  DCNTD12 | GRD11        | GRD11      | GRD11  — — — — — — — — — — — — — — — — — — | GRD11       | GRD11  GRD12  GRD13  GR | GRD11  GRD12  GRD13  GRD13  GRD13  GRD13  GRD13  GRD13  GRD14  DCNTD10  DCNTD10  DCNTD10  DCNTD11  DCNTD11  DCNTD11  DCNTD12  TIOR1D2  OSSD23[1:0]  OSSD22[1:0]  OSSD21[1:0] | GRD12  GRD12  GRD13  GRD13  DCNTD10  DCNTD10  DCNTD11  DCNTD11  DCNTD12  TIOR1D2  OSSD23[1:0]  OSSD21[1:0]  OSSD21[1:0]  OSSD21[1:0] |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5  | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1   | 24/16/8/0 |
|---------|------------------|------------|------------|-------------|------------|------------|------------|-------------|-----------|
| ATU-III | TIOR2D2          | _          |            | IOBD23[2:0] | ]          |            |            | IOBD22[2:0] | ]         |
|         |                  |            |            | IOBD21[2:0] | ]          | _          |            | IOBD20[2:0  | ]         |
|         | DSTRD2           | _          | _          |             | _          | DSTD23     | DSTD22     | DSTD21      | DSTD20    |
|         | DSRD2            | _          | _          | _           | _          | DSFD23     | DSFD22     | DSFD21      | DSFD20    |
|         | DCRD2            | _          | TF         | RGSELD23[2  | 2:0]       | _          | TF         | RGSELD22[2  | 2:0]      |
|         |                  | _          | TF         | RGSELD21[2  | 2:0]       |            | TF         | RGSELD20[2  | 2:0]      |
|         | TSRD2            | _          | _          | OVF2D2      | OVF1D2     | UDFD23     | UDFD22     | UDFD21      | UDFD20    |
|         |                  | CMFAD23    | CMFAD22    | CMFAD21     | CMFAD20    | CMFBD23    | CMFBD22    | CMFBD21     | CMFBD20   |
|         | TIERD2           | _          | _          | OVE2D2      | OVE1D2     | UDED23     | UDED22     | UDED21      | UDED20    |
|         |                  | CMEAD23    | CMEAD22    | CMEAD21     | CMEAD20    | CMEBD23    | CMEBD22    | CMEBD21     | CMEBD20   |
|         | OCRD20           |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  | _          | _          | _           |            |            | _          | _           | _         |
|         | OCRD21           |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  | _          | _          | _           |            |            | _          | _           | _         |
|         | OCRD22           |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  | _          | _          |             | _          | _          |            | _           | _         |
|         | OCRD23           |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  | _          | _          | _           |            |            | _          | _           | _         |
|         | GRD20            |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  |            |            |             |            |            |            |             |           |
|         |                  |            |            | <u> </u>    | <u> </u>   | <u> </u>   | <u> </u>   | <u></u>     | <u> </u>  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|---------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| ATU-III | GRD21            |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            | _          | _          | _          | _          | _          | _         | _         |
|         | GRD22            |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  | _          | _          |            | _          |            |            |           | _         |
|         | GRD23            |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            | _          | _          | _          |            | _          | _         |           |
|         | DCNTD20          |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  | _          | _          | _          | _          | _          | _          | _         | _         |
|         | DCNTD21          |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  | _          | _          | _          | _          |            | _          | _         |           |
|         | DCNTD22          |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  | _          | _          |            | _          |            |            |           | _         |
|         | DCNTD23          |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  | _          | _          | _          | _          |            | _          | _         | _         |

RENESAS

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5  | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|---------|------------------|------------|------------|-------------|------------|------------|------------|------------|-----------|
| ATU-III | TIOR1D3          | OSSD       | 33[1:0]    | OSSD        | 32[1:0]    | OSSD       | 31[1:0]    | OSSD       | 30[1:0]   |
|         |                  | IOAD       | 33[1:0]    | IOAD:       | 32[1:0]    | IOAD:      | 31[1:0]    | IOAD       | 30[1:0]   |
|         | TIOR2D3          | _          |            | IOBD33[2:0] | ]          |            |            | IOBD32[2:0 | ]         |
|         |                  | _          |            | IOBD31[2:0] | ]          | _          |            | IOBD30[2:0 | ]         |
|         | DSTRD3           |            | _          |             |            | DSTD33     | DSTD32     | DSTD31     | DSTD30    |
|         | DSRD3            | _          | _          | _           | _          | DSFD33     | DSFD32     | DSFD31     | DSFD30    |
|         | DCRD3            |            | TF         | RGSELD33[2  | 2:0]       | _          | TF         | RGSELD32[2 | 2:0]      |
|         |                  | _          | TF         | RGSELD31[2  | 2:0]       |            | TF         | RGSELD30[2 | 2:0]      |
|         | TSRD3            | _          | _          | OVF2D3      | OVF1D3     | UDFD33     | UDFD32     | UDFD31     | UDFD30    |
|         |                  | CMFAD33    | CMFAD32    | CMFAD31     | CMFAD30    | CMFBD33    | CMFBD32    | CMFBD31    | CMFBD30   |
|         | TIERD3           | _          | _          | OVE2D3      | OVE1D3     | UDED33     | UDED32     | UDED31     | UDED30    |
|         |                  | CMEAD33    | CMEAD32    | CMEAD31     | CMEAD30    | CMEBD33    | CMEBD32    | CMEBD31    | CMEBD30   |
|         | OCRD30           |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            | _          |             | _          |            |            | _          | _         |
|         | OCRD31           |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  | _          | _          | _           |            |            |            |            | —         |
|         | OCRD32           |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  | _          | _          | _           | _          | _          | _          | _          | _         |
|         | OCRD33           |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            |           |
|         |                  |            |            |             |            |            |            |            | _         |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|---------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| ATU-III | GRD30            |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  | _          | _          | _          | _          |            |            |           |           |
|         | GRD31            |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  | _          | _          | _          | _          | _          | _          | _         | _         |
|         | GRD32            |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            | _          | _          | _          | _          | _          | _         | _         |
|         | GRD33            |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            | _          | _          | _          |            | _         |           |
|         | DCNTD30          |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  | _          |            | _          | _          | _          | _          | _         | _         |
|         | DCNTD31          |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  | _          | _          | _          | _          | _          | _          | _         | _         |
|         | DCNTD32          |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            |            |            |            |            |            |           |           |
|         |                  |            | _          |            | _          |            | _          |           | _         |

RENESAS

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6    | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |  |  |  |  |
|---------|------------------|---------------|---------------|------------|------------|------------|------------|------------|-----------|--|--|--|--|
| ATU-III | DCNTD33          |               |               |            |            |            |            |            |           |  |  |  |  |
|         |                  |               |               |            |            |            |            |            |           |  |  |  |  |
|         |                  |               |               |            |            |            |            |            |           |  |  |  |  |
|         |                  | _             | _             | _          | _          | _          | _          | _          | _         |  |  |  |  |
|         | TSTRE            |               | _             | STRE5      | STRE4      | STRE3      | STRE2      | STRE1      | STRE0     |  |  |  |  |
|         | TCRE0            |               |               |            | _          |            | C          | KSELE0[2:0 | D]        |  |  |  |  |
|         | TOCRE0           |               | _             | _          | _          | TONEE03    | TONEE02    | TONEE01    | TONEE00   |  |  |  |  |
|         | TIERE0           |               | _             | _          | _          | CMEE03     | CMEE02     | CMEE01     | CMEE00    |  |  |  |  |
|         | RLDCRE0          | _             | _             | _          | _          | RLDENE03   | RLDENE02   | RLDENE01   | RLDENE00  |  |  |  |  |
|         | TSRE0            | OVFE03        | OVFE02        | OVFE01     | OVFE00     | CMFE03     | CMFE02     | CMFE01     | CMFE00    |  |  |  |  |
|         | PSCRE0           |               | _             |            |            | _          |            | PSCE0[2:0] |           |  |  |  |  |
|         | SSTRE0           | _             | _             | _          | _          | SSTRE03    | SSTRE02    |            |           |  |  |  |  |
|         | CYLRE00          |               | CYLRE00[15:8] |            |            |            |            |            |           |  |  |  |  |
|         |                  |               | CYLRE00[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | CYLRE01          | CYLRE01[15:8] |               |            |            |            |            |            |           |  |  |  |  |
|         |                  |               | CYLRE01[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | CYLRE02          |               | CYLRE02[15:8] |            |            |            |            |            |           |  |  |  |  |
|         |                  |               |               |            | CYLRE      | [02[7:0]   |            |            |           |  |  |  |  |
|         | CYLRE03          |               |               |            | CYLRE      | 03[15:8]   |            |            |           |  |  |  |  |
|         |                  |               |               |            | CYLRE      | [03[7:0]   |            |            |           |  |  |  |  |
|         | DTRE00           |               |               |            | DTREC      | 0[15:8]    |            |            |           |  |  |  |  |
|         |                  |               |               |            | DTRE       | 00[7:0]    |            |            |           |  |  |  |  |
|         | DTRE01           |               |               |            | DTREC      | 1[15:8]    |            |            |           |  |  |  |  |
|         |                  |               |               |            | DTRE       | 01[7:0]    |            |            |           |  |  |  |  |
|         | DTRE02           |               |               |            | DTREC      | 2[15:8]    |            |            |           |  |  |  |  |
|         |                  |               |               |            | DTRE       | 02[7:0]    |            |            |           |  |  |  |  |
|         | DTRE03           |               |               |            | DTREC      | 3[15:8]    |            |            |           |  |  |  |  |
|         |                  | DTRE03[7:0]   |               |            |            |            |            |            |           |  |  |  |  |
|         | CRLDE00          |               |               |            | CRLDE      | 00[15:8]   |            |            |           |  |  |  |  |
|         |                  |               |               |            | CRLDE      | 00[7:0]    |            |            |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6    | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1   | 24/16/8/0 |  |  |  |  |
|---------|------------------|------------|---------------|------------|------------|------------|------------|-------------|-----------|--|--|--|--|
| ATU-III | CRLDE01          |            | I.            |            | CRLDE      | 01[15:8]   | I .        |             |           |  |  |  |  |
|         |                  |            |               |            | CRLDE      | E01[7:0]   |            |             |           |  |  |  |  |
|         | CRLDE02          |            |               |            | CRLDE      | 02[15:8]   |            |             |           |  |  |  |  |
|         |                  |            |               |            | CRLDE      | 02[7:0]    |            |             |           |  |  |  |  |
|         | CRLDE03          |            |               |            | CRLDE      | 03[15:8]   |            |             |           |  |  |  |  |
|         |                  |            |               |            | CRLDE      | [03[7:0]   |            |             |           |  |  |  |  |
|         | DRLDE00          |            |               |            | DRLDE      | 00[15:8]   |            |             |           |  |  |  |  |
|         |                  |            |               |            | DRLDE      | 00[7:0]    |            |             |           |  |  |  |  |
|         | DRLDE01          |            |               |            | DRLDE      | 01[15:8]   |            |             |           |  |  |  |  |
|         |                  |            |               |            | DRLDE      | 01[7:0]    |            |             |           |  |  |  |  |
|         | DRLDE02          |            |               |            | DRLDE      | 02[15:8]   |            |             |           |  |  |  |  |
|         |                  |            | DRLDE02[7:0]  |            |            |            |            |             |           |  |  |  |  |
|         | DRLDE03          |            | DRLDE03[15:8] |            |            |            |            |             |           |  |  |  |  |
|         |                  |            | DRLDE03[7:0]  |            |            |            |            |             |           |  |  |  |  |
|         | TCNTE00          |            | TCNTE00[15:8] |            |            |            |            |             |           |  |  |  |  |
|         |                  |            |               |            |            | [00[7:0]   |            |             |           |  |  |  |  |
|         | TCNTE01          |            |               |            |            | 01[15:8]   |            |             |           |  |  |  |  |
|         |                  |            |               |            |            | [01[7:0]   |            |             |           |  |  |  |  |
|         | TCNTE02          |            |               |            |            | 02[15:8]   |            |             |           |  |  |  |  |
|         | T01/TF00         |            |               |            |            | :02[7:0]   |            |             |           |  |  |  |  |
|         | TCNTE03          |            |               |            |            | 03[15:8]   |            |             |           |  |  |  |  |
|         | TODE             |            | 1             |            | ICNIE      | [03[7:0]   |            | OKOEL E410. | 21        |  |  |  |  |
|         | TCRE1 TOCRE1     | -          |               |            |            | TONEE13    | TONEE12    | TONEE11     | TONEE10   |  |  |  |  |
|         | TIERE1           |            |               |            |            | CMEE13     | CMEE12     | CMEE11      | CMEE10    |  |  |  |  |
|         | RLDCRE1          |            |               |            |            | RLDENE13   | RLDENE12   |             | RLDENE10  |  |  |  |  |
|         | TSRE1            | OVFE13     | OVFE12        | OVFE11     | OVFE10     | CMFE13     | CMFE12     | CMFE11      | CMFE10    |  |  |  |  |
|         | PSCRE1           | _          |               |            |            |            | J.III E 12 | PSCE1[2:0]  | <u> </u>  |  |  |  |  |
|         |                  |            | _             |            |            | SSTRE13    | SSTRE12    |             | SSTRE10   |  |  |  |  |
|         | 0011121          | SSTRE1     |               |            |            |            |            |             |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6                    | 29/21/13/5 | 28/20/12/4         | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|---------------|-------------------------------|------------|--------------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | CYLRE10          |               |                               | •          | CYLRE.             | 10[15:8]   |            |           |           |  |  |  |  |
|         |                  |               |                               |            | CYLRE              | 10[7:0]    |            |           |           |  |  |  |  |
|         | CYLRE11          |               |                               |            | CYLRE <sup>-</sup> | 11[15:8]   |            |           |           |  |  |  |  |
|         |                  |               | CYLRE11[7:0]<br>CYLRE12[15:8] |            |                    |            |            |           |           |  |  |  |  |
|         | CYLRE12          |               |                               |            |                    |            |            |           |           |  |  |  |  |
|         |                  |               |                               |            | CYLRE              | 12[7:0]    |            |           |           |  |  |  |  |
|         | CYLRE13          |               |                               |            | CYLRE*             | 13[15:8]   |            |           |           |  |  |  |  |
|         |                  |               | CYLRE13[7:0]                  |            |                    |            |            |           |           |  |  |  |  |
|         | DTRE10           |               | DTRE10[15:8]                  |            |                    |            |            |           |           |  |  |  |  |
|         |                  |               | DTRE10[7:0]                   |            |                    |            |            |           |           |  |  |  |  |
|         | DTRE11           |               | DTRE11[15:8]                  |            |                    |            |            |           |           |  |  |  |  |
|         |                  |               |                               |            | DTRE               | 11[7:0]    |            |           |           |  |  |  |  |
|         | DTRE12           |               | DTRE12[15:8] DTRE12[7:0]      |            |                    |            |            |           |           |  |  |  |  |
|         |                  |               |                               |            |                    |            |            |           |           |  |  |  |  |
|         | DTRE13           |               |                               |            | DTRE1              | 3[15:8]    |            |           |           |  |  |  |  |
|         |                  |               |                               |            | DTRE               | 13[7:0]    |            |           |           |  |  |  |  |
|         | CRLDE10          |               |                               |            | CRLDE <sup>-</sup> | 10[15:8]   |            |           |           |  |  |  |  |
|         |                  |               |                               |            | CRLDE              | 10[7:0]    |            |           |           |  |  |  |  |
|         | CRLDE11          |               |                               |            | CRLDE <sup>-</sup> | 11[15:8]   |            |           |           |  |  |  |  |
|         |                  |               |                               |            | CRLDE              | 11[7:0]    |            |           |           |  |  |  |  |
|         | CRLDE12          |               |                               |            | CRLDE <sup>-</sup> | 12[15:8]   |            |           |           |  |  |  |  |
|         |                  |               |                               |            | CRLDE              | 12[7:0]    |            |           |           |  |  |  |  |
|         | CRLDE13          |               |                               |            | CRLDE <sup>-</sup> | 13[15:8]   |            |           |           |  |  |  |  |
|         |                  |               |                               |            | CRLDE              | 13[7:0]    |            |           |           |  |  |  |  |
|         | DRLDE10          |               |                               |            | DRLDE <sup>-</sup> | 10[15:8]   |            |           |           |  |  |  |  |
|         |                  |               |                               |            | DRLDE              | 10[7:0]    |            |           |           |  |  |  |  |
|         | DRLDE11          | DRLDE11[15:8] |                               |            |                    |            |            |           |           |  |  |  |  |
|         |                  |               | DRLDE11[7:0]                  |            |                    |            |            |           |           |  |  |  |  |
|         | DRLDE12          | DRLDE12[15:8] |                               |            |                    |            |            |           |           |  |  |  |  |
|         |                  |               |                               |            | DRLDE              | 12[7:0]    |            |           |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2           | 25/17/9/1  | 24/16/8/0 |  |  |
|---------|------------------|---------------|--------------|------------|------------|------------|----------------------|------------|-----------|--|--|
| ATU-III | DRLDE13          |               | I            | I          | DRLDE      | 13[15:8]   |                      | I          |           |  |  |
|         |                  |               |              |            | DRLDE      | 13[7:0]    |                      |            |           |  |  |
|         | TCNTE10          |               |              |            | TCNTE      | 10[15:8]   |                      |            |           |  |  |
|         |                  |               |              |            | TCNTE      | 10[7:0]    |                      |            |           |  |  |
|         | TCNTE11          |               |              |            | TCNTE      | 11[15:8]   |                      |            |           |  |  |
|         |                  |               |              |            | TCNTE      | 11[7:0]    |                      |            |           |  |  |
|         | TCNTE12          |               |              |            | TCNTE      | 12[15:8]   |                      |            |           |  |  |
|         |                  |               |              |            | TCNTE      | 12[7:0]    |                      |            |           |  |  |
|         | TCNTE13          | TCNTE13[15:8] |              |            |            |            |                      |            |           |  |  |
|         |                  |               | TCNTE13[7:0] |            |            |            |                      |            |           |  |  |
|         | TCRE2            | _             | _            | _          | _          | _          | C                    | KSELE2[2:  | 0]        |  |  |
|         | TOCRE2           | _             | _            | _          | _          | TONEE23    | TONEE22              | TONEE21    | TONEE20   |  |  |
|         | TIERE2           | _             | _            | _          | _          | CMEE23     | CMEE22               | CMEE21     | CMEE20    |  |  |
|         | RLDCRE2          | _             | _            | _          | _          | RLDENE23   | RLDENE22             | RLDENE21   | RLDENE20  |  |  |
|         | TSRE2            | OVFE23        | OVFE22       | OVFE21     | OVFE20     | CMFE23     | CMFE22 CMFE21 CMFE20 |            |           |  |  |
|         | PSCRE2           | _             | _            | _          | _          | _          |                      | PSCE2[2:0] |           |  |  |
|         | SSTRE2           | _             |              | _          | _          | SSTRE23    | SSTRE22              | SSTRE21    | SSTRE20   |  |  |
|         | CYLRE20          |               |              |            | CYLRE      | 20[15:8]   |                      |            |           |  |  |
|         |                  |               |              |            | CYLRE      | 20[7:0]    |                      |            |           |  |  |
|         | CYLRE21          |               |              |            | CYLRE      | 21[15:8]   |                      |            |           |  |  |
|         |                  |               |              |            | CYLRE      | 21[7:0]    |                      |            |           |  |  |
|         | CYLRE22          |               |              |            | CYLRE      | 22[15:8]   |                      |            |           |  |  |
|         |                  |               |              |            | CYLRE      | 22[7:0]    |                      |            |           |  |  |
|         | CYLRE23          |               |              |            | CYLRE      | 23[15:8]   |                      |            |           |  |  |
|         |                  |               |              |            | CYLRE      | 23[7:0]    |                      |            |           |  |  |
|         | DTRE20           |               |              |            | DTRE2      | 20[15:8]   |                      |            |           |  |  |
|         | DTRE20[7:0]      |               |              |            |            |            |                      |            |           |  |  |
|         | DTRE21           |               |              |            | DTRE2      | 21[15:8]   |                      |            |           |  |  |
|         |                  |               |              |            | DTRE       | 21[7:0]    |                      |            |           |  |  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6    | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |  |  |  |  |
|---------|------------------|------------|---------------|------------|------------|------------|------------|------------|-----------|--|--|--|--|
| ATU-III | DTRE22           |            | <u>I</u>      | <u>I</u>   | DTRE2      | 22[15:8]   | ı          |            |           |  |  |  |  |
|         |                  |            |               |            | DTRE       | 22[7:0]    |            |            |           |  |  |  |  |
|         | DTRE23           |            |               |            | DTRE2      | 23[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |               |            | DTRE       | 23[7:0]    |            |            |           |  |  |  |  |
|         | CRLDE20          |            |               |            | CRLDE      | 20[15:8]   |            |            |           |  |  |  |  |
|         |                  |            | CRLDE20[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | CRLDE21          |            | CRLDE21[15:8] |            |            |            |            |            |           |  |  |  |  |
|         |                  |            | CRLDE21[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | CRLDE22          |            |               |            | CRLDE      | 22[15:8]   |            |            |           |  |  |  |  |
|         |                  |            | CRLDE22[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | CRLDE23          |            |               |            | CRLDE      | 23[15:8]   |            |            |           |  |  |  |  |
|         |                  |            | CRLDE23[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | DRLDE20          |            |               |            | DRLDE      | 20[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |               |            |            |            |            |            |           |  |  |  |  |
|         | DRLDE21          |            |               |            | DRLDE      | 21[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |               |            | DRLDE      | 21[7:0]    |            |            |           |  |  |  |  |
|         | DRLDE22          |            |               |            | DRLDE      | 22[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |               |            | DRLDE      | 22[7:0]    |            |            |           |  |  |  |  |
|         | DRLDE23          |            |               |            | DRLDE      | 23[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |               |            | DRLDE      | [23[7:0]   |            |            |           |  |  |  |  |
|         | TCNTE20          |            |               |            | TCNTE      | 20[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |               |            | TCNTE      | 20[7:0]    |            |            |           |  |  |  |  |
|         | TCNTE21          |            |               |            | TCNTE      | 21[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |               |            | TCNTE      | 21[7:0]    |            |            |           |  |  |  |  |
|         | TCNTE22          |            |               |            | TCNTE      | 22[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |               |            | TCNTE      | [22[7:0]   |            |            |           |  |  |  |  |
|         | TCNTE23          |            | TCNTE23[15:8] |            |            |            |            |            |           |  |  |  |  |
|         |                  |            | TCNTE23[7:0]  |            |            |            |            |            |           |  |  |  |  |
|         | TCRE3            | _          |               | _          | _          | _          | C          | CKSELE3[2: | 0]        |  |  |  |  |
|         | TOCRE3           |            |               |            |            | TONEE33    | TONEE32    | TONEE31    | TONEE30   |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |  |  |  |  |
|---------|------------------|--------------|--------------|------------|------------|------------|------------|------------|-----------|--|--|--|--|
| ATU-III | TIERE3           | _            | _            | _          | _          | CMEE33     | CMEE32     | CMEE31     | CMEE30    |  |  |  |  |
|         | RLDCRE3          |              |              |            | _          | RLDENE33   | RLDENE32   | RLDENE31   | RLDENE30  |  |  |  |  |
|         | TSRE3            | OVFE33       | OVFE32       | OVFE31     | OVFE30     | CMFE33     | CMFE32     | CMFE31     | CMFE30    |  |  |  |  |
|         | PSCRE3           | _            | _            | _          | _          | _          |            | PSCE3[2:0] | •         |  |  |  |  |
|         | SSTRE3           | _            | _            | _          | _          | SSTRE33    | SSTRE32    | SSTRE31    | SSTRE30   |  |  |  |  |
|         | CYLRE30          |              |              |            | CYLRE      | 30[15:8]   |            | 1          |           |  |  |  |  |
|         |                  |              |              |            | CYLRE      | 30[7:0]    |            |            |           |  |  |  |  |
|         | CYLRE31          |              |              |            | CYLRE      | 31[15:8]   |            |            |           |  |  |  |  |
|         |                  |              |              |            | CYLRE      | 31[7:0]    |            |            |           |  |  |  |  |
|         | CYLRE32          |              |              |            | CYLRE      | 32[15:8]   |            |            |           |  |  |  |  |
|         |                  |              |              |            | CYLRE      | 32[7:0]    |            |            |           |  |  |  |  |
|         | CYLRE33          |              |              |            | CYLRE      | 33[15:8]   |            |            |           |  |  |  |  |
|         |                  |              | CYLRE33[7:0] |            |            |            |            |            |           |  |  |  |  |
|         | DTRE30           |              | DTRE30[15:8] |            |            |            |            |            |           |  |  |  |  |
|         |                  |              |              |            | DTRE       | 30[7:0]    |            |            |           |  |  |  |  |
|         | DTRE31           |              |              |            | DTRE3      | 1[15:8]    |            |            |           |  |  |  |  |
|         |                  |              |              |            | DTRE       |            |            |            |           |  |  |  |  |
|         | DTRE32           |              |              |            | DTRE3      |            |            |            |           |  |  |  |  |
|         |                  |              |              |            | DTRE       | 32[7:0]    |            |            |           |  |  |  |  |
|         | DTRE33           |              |              |            | DTRE3      |            |            |            |           |  |  |  |  |
|         |                  |              |              |            | DTRE       | 33[7:0]    |            |            |           |  |  |  |  |
|         | CRLDE30          |              |              |            | CRLDE      |            |            |            |           |  |  |  |  |
|         |                  |              |              |            | CRLDE      | 30[7:0]    |            |            |           |  |  |  |  |
|         | CRLDE31          |              |              |            | CRLDE      |            |            |            |           |  |  |  |  |
|         |                  |              |              |            | CRLDE      |            |            |            |           |  |  |  |  |
|         | CRLDE32          |              |              |            | CRLDE      |            |            |            |           |  |  |  |  |
|         |                  | CRLDE32[7:0] |              |            |            |            |            |            |           |  |  |  |  |
|         | CRLDE33          |              |              |            | CRLDE      |            |            |            |           |  |  |  |  |
|         |                  |              |              |            | CRLDE      | 33[7:0]    |            |            |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6                 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |  |  |  |  |
|---------|------------------|------------|----------------------------|------------|------------|------------|------------|------------|-----------|--|--|--|--|
| ATU-III | DRLDE30          |            | •                          | •          | DRLDE      | 30[15:8]   |            |            | •         |  |  |  |  |
|         |                  |            |                            |            | DRLDE      | 30[7:0]    |            |            |           |  |  |  |  |
|         | DRLDE31          |            |                            |            | DRLDE      | 31[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |                            |            | DRLDE      | 31[7:0]    |            |            |           |  |  |  |  |
|         | DRLDE32          |            |                            |            | DRLDE      | 32[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |                            |            | DRLDE      | 32[7:0]    |            |            |           |  |  |  |  |
|         | DRLDE33          |            |                            |            | DRLDE      | 33[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |                            |            | DRLDE      | 33[7:0]    |            |            |           |  |  |  |  |
|         | TCNTE30          |            |                            |            | TCNTE      | 30[15:8]   |            |            |           |  |  |  |  |
|         |                  |            | TCNTE30[7:0]               |            |            |            |            |            |           |  |  |  |  |
|         | TCNTE31          |            | TCNTE31[15:8]              |            |            |            |            |            |           |  |  |  |  |
|         |                  |            | TCNTE31[7:0]               |            |            |            |            |            |           |  |  |  |  |
|         | TCNTE32          |            | TCNTE32[15:8] TCNTE32[7:0] |            |            |            |            |            |           |  |  |  |  |
|         |                  |            |                            |            |            |            |            |            |           |  |  |  |  |
|         | TCNTE33          |            | TCNTE33[15:8]              |            |            |            |            |            |           |  |  |  |  |
|         |                  |            |                            |            | TCNTE      | 33[7:0]    |            |            |           |  |  |  |  |
|         | TCRE4            | _          |                            | _          | _          | _          | C          | KSELE4[2:0 | )]        |  |  |  |  |
|         | TOCRE4           | _          |                            | _          | —          | TONEE43    | TONEE42    | TONEE41    | TONEE40   |  |  |  |  |
|         | TIERE4           | _          |                            | _          | _          | CMEE43     | CMEE42     | CMEE41     | CMEE40    |  |  |  |  |
|         | RLDCRE4          | _          |                            | _          | _          | RLDENE43   | RLDENE42   | RLDENE41   | RLDENE40  |  |  |  |  |
|         | TSRE4            | OVFE43     | OVFE42                     | OVFE41     | OVFE40     | CMFE43     | CMFE42     | CMFE41     | CMFE40    |  |  |  |  |
|         | PSCRE4           | _          |                            | _          | _          | _          |            | PSCE4[2:0] |           |  |  |  |  |
|         | SSTRE4           | _          |                            | _          | _          | SSTRE43    | SSTRE42    | SSTRE41    | SSTRE40   |  |  |  |  |
|         | CYLRE40          |            |                            |            | CYLRE      | 40[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |                            |            | CYLRE      | 40[7:0]    |            |            |           |  |  |  |  |
|         | CYLRE41          |            |                            |            | CYLRE      | 41[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |                            |            |            |            |            |            |           |  |  |  |  |
|         | CYLRE42          |            |                            |            | CYLRE      | 42[15:8]   |            |            |           |  |  |  |  |
|         |                  |            |                            |            | CYLRE      | 42[7:0]    |            |            |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |  |
|---------|------------------|---------------|--------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|--|
| ATU-III | CYLRE43          |               |              |            | CYLRE4     | 13[15:8]   | I          |           |           |  |  |  |  |  |
|         |                  |               |              |            | CYLRE      | 43[7:0]    |            |           |           |  |  |  |  |  |
|         | DTRE40           |               |              |            | DTRE4      | 0[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | DTRE4      | 10[7:0]    |            |           |           |  |  |  |  |  |
|         | DTRE41           |               | DTRE41[15:8] |            |            |            |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | DTRE4      | 11[7:0]    |            |           |           |  |  |  |  |  |
|         | DTRE42           |               |              |            | DTRE4      | 2[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |               | DTRE42[7:0]  |            |            |            |            |           |           |  |  |  |  |  |
|         | DTRE43           |               |              |            | DTRE4      | 3[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | DTRE4      | 13[7:0]    |            |           |           |  |  |  |  |  |
|         | CRLDE40          |               |              |            | CRLDE4     | 40[15:8]   |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | CRLDE      | 40[7:0]    |            |           |           |  |  |  |  |  |
|         | CRLDE41          |               |              |            | CRLDE4     | 41[15:8]   |            |           |           |  |  |  |  |  |
|         |                  |               | CRLDE41[7:0] |            |            |            |            |           |           |  |  |  |  |  |
|         | CRLDE42          |               |              |            | CRLDE4     | 12[15:8]   |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | CRLDE      | 42[7:0]    |            |           |           |  |  |  |  |  |
|         | CRLDE43          |               |              |            | CRLDE4     | 43[15:8]   |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | CRLDE      | 43[7:0]    |            |           |           |  |  |  |  |  |
|         | DRLDE40          |               |              |            | DRLDE      | 40[15:8]   |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | DRLDE      | 40[7:0]    |            |           |           |  |  |  |  |  |
|         | DRLDE41          |               |              |            | DRLDE      | 41[15:8]   |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | DRLDE      | 41[7:0]    |            |           |           |  |  |  |  |  |
|         | DRLDE42          |               |              |            | DRLDE      | 12[15:8]   |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | DRLDE      | 42[7:0]    |            |           |           |  |  |  |  |  |
|         | DRLDE43          |               |              |            | DRLDE4     | 43[15:8]   |            |           |           |  |  |  |  |  |
|         |                  | DRLDE43[7:0]  |              |            |            |            |            |           |           |  |  |  |  |  |
|         | TCNTE40          | TCNTE40[15:8] |              |            |            |            |            |           |           |  |  |  |  |  |
|         |                  | TCNTE40[7:0]  |              |            |            |            |            |           |           |  |  |  |  |  |
|         | TCNTE41          |               |              |            | TCNTE      |            |            |           |           |  |  |  |  |  |
|         |                  |               |              |            | TCNTE      | 41[7:0]    |            |           |           |  |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7          | 30/22/14/6                  | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |  |  |  |  |  |
|---------|------------------|---------------------|-----------------------------|------------|------------|------------|------------|------------|-----------|--|--|--|--|--|
| ATU-III | TCNTE42          |                     | L                           | ·          | TCNTE      | 42[15:8]   | Į.         | Į.         | ı         |  |  |  |  |  |
|         |                  |                     |                             |            | TCNTE      | 42[7:0]    |            |            |           |  |  |  |  |  |
|         | TCNTE43          |                     |                             |            | TCNTE      | 43[15:8]   |            |            |           |  |  |  |  |  |
|         |                  | TCNTE43[7:0]        |                             |            |            |            |            |            |           |  |  |  |  |  |
|         | TCRE5            | TCRE5 — — — — CKSEL |                             |            |            |            |            |            |           |  |  |  |  |  |
|         | TOCRE5           | _                   | _                           | _          | _          | TONEE53    | TONEE52    | TONEE51    | TONEE50   |  |  |  |  |  |
|         | TIERE5           |                     | _                           | _          | _          | CMEE53     | CMEE52     | CMEE51     | CMEE50    |  |  |  |  |  |
|         | RLDCRE5          |                     |                             |            | _          | RLDENE53   | RLDENE52   | RLDENE51   | RLDENE50  |  |  |  |  |  |
|         | TSRE5            | OVFE53              | OVFE52                      | OVFE51     | OVFE50     | CMFE53     | CMFE52     | CMFE51     | CMFE50    |  |  |  |  |  |
|         | PSCRE5           |                     | _                           | _          | _          | _          |            | PSCE5[2:0] |           |  |  |  |  |  |
|         | SSTRE5           |                     | _                           | _          | _          | SSTRE53    | SSTRE52    | SSTRE51    | SSTRE50   |  |  |  |  |  |
|         | CYLRE50          |                     |                             |            | CYLRE      | 50[15:8]   |            |            |           |  |  |  |  |  |
|         |                  |                     | CYLRE50[7:0]                |            |            |            |            |            |           |  |  |  |  |  |
|         | CYLRE51          |                     | CYLRE51[15:8]               |            |            |            |            |            |           |  |  |  |  |  |
|         |                  |                     |                             |            |            |            |            |            |           |  |  |  |  |  |
|         | CYLRE52          |                     | CYLRE52[15:8]  CYLRE52[7:0] |            |            |            |            |            |           |  |  |  |  |  |
|         |                  |                     |                             |            |            |            |            |            |           |  |  |  |  |  |
|         | CYLRE53          |                     | CYLRE53[15:8]               |            |            |            |            |            |           |  |  |  |  |  |
|         |                  |                     |                             |            | CYLRE      | 53[7:0]    |            |            |           |  |  |  |  |  |
|         | DTRE50           |                     |                             |            | DTRE5      | 60[15:8]   |            |            |           |  |  |  |  |  |
|         |                  |                     |                             |            | DTRE       | 50[7:0]    |            |            |           |  |  |  |  |  |
|         | DTRE51           |                     |                             |            | DTRE5      | 51[15:8]   |            |            |           |  |  |  |  |  |
|         |                  |                     |                             |            | DTRE       | 51[7:0]    |            |            |           |  |  |  |  |  |
|         | DTRE52           |                     |                             |            | DTRE5      | 52[15:8]   |            |            |           |  |  |  |  |  |
|         |                  |                     |                             |            | DTRE       | 52[7:0]    |            |            |           |  |  |  |  |  |
|         | DTRE53           |                     |                             |            | DTRE5      | 3[15:8]    |            |            |           |  |  |  |  |  |
|         |                  |                     |                             |            |            |            |            |            |           |  |  |  |  |  |
|         | CRLDE50          |                     |                             |            | CRLDE      | 50[15:8]   |            |            |           |  |  |  |  |  |
|         |                  |                     |                             |            | CRLDE      | 50[7:0]    |            |            |           |  |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6                  | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|--------------|-----------------------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | CRLDE51          |              |                             |            | CRLDE      | 51[15:8]   |            |           | 1         |  |  |  |  |
|         |                  |              |                             |            | CRLDE      | 51[7:0]    |            |           |           |  |  |  |  |
|         | CRLDE52          |              |                             |            | CRLDE      | 52[15:8]   |            |           |           |  |  |  |  |
|         |                  |              |                             |            | CRLDE      | 52[7:0]    |            |           |           |  |  |  |  |
|         | CRLDE53          |              |                             |            | CRLDE      | 53[15:8]   |            |           |           |  |  |  |  |
|         |                  | CRLDE53[7:0] |                             |            |            |            |            |           |           |  |  |  |  |
|         | DRLDE50          |              | DRLDE50[15:8]               |            |            |            |            |           |           |  |  |  |  |
|         |                  |              |                             |            | DRLDE      | 50[7:0]    |            |           |           |  |  |  |  |
|         | DRLDE51          |              |                             |            | DRLDE      | 51[15:8]   |            |           |           |  |  |  |  |
|         |                  |              | DRLDE51[7:0]                |            |            |            |            |           |           |  |  |  |  |
|         | DRLDE52          |              | DRLDE52[15:8]               |            |            |            |            |           |           |  |  |  |  |
|         |                  |              | DRLDE52[7:0]                |            |            |            |            |           |           |  |  |  |  |
|         | DRLDE53          |              | DRLDE53[15:8]  DRLDE53[7:0] |            |            |            |            |           |           |  |  |  |  |
|         |                  |              |                             |            |            |            |            |           |           |  |  |  |  |
|         | TCNTE50          |              | TCNTE50[15:8]               |            |            |            |            |           |           |  |  |  |  |
|         |                  |              | TCNTE50[7:0]                |            |            |            |            |           |           |  |  |  |  |
|         | TCNTE51          |              |                             |            | TCNTE      | 51[15:8]   |            |           |           |  |  |  |  |
|         |                  |              |                             |            | TCNTE      | 51[7:0]    |            |           |           |  |  |  |  |
|         | TCNTE52          |              |                             |            | TCNTE      | 52[15:8]   |            |           |           |  |  |  |  |
|         |                  |              |                             |            | TCNTE      | 52[7:0]    |            |           |           |  |  |  |  |
|         | TCNTE53          |              |                             |            | TCNTE      | 53[15:8]   |            |           |           |  |  |  |  |
|         |                  |              |                             |            | TCNTE      | 53[7:0]    |            |           |           |  |  |  |  |
|         | TSTRF            |              | _                           |            | _          | _          | _          | —         | _         |  |  |  |  |
|         |                  | _            | _                           | _          | _          | STRF19     | STRF18     | STRF17    | STRF16    |  |  |  |  |
|         |                  | STRF15       | STRF14                      | STRF13     | STRF12     | STRF11     | STRF10     | STRF9     | STRF8     |  |  |  |  |
|         |                  | STRF7        | STRF6                       | STRF5      | STRF4      | STRF3      | STRF2      | STRF1     | STRF0     |  |  |  |  |
|         | NCCRF            | _            | _                           | _          | _          | _          | _          | _         | _         |  |  |  |  |
|         |                  | _            | _                           | _          |            | NCEF19     | NCEF18     | NCEF17    | NCEF16    |  |  |  |  |
|         |                  | NCEF15       | NCEF14                      | NCEF13     | NCEF12     | NCEF11     | NCEF10     | NCEF9     | NCEF8     |  |  |  |  |
|         |                  | NCEF7        | NCEF6                       | NCEF5      | NCEF4      | NCEF3      | NCEF2      | NCEF1     | NCEF0     |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |  |
|---------|------------------|--------------|--------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|--|
| ATU-III | NCNTFA0          |              |              | 1          | NCNTF.     | A0[7:0]    | ı          | 1         | 1         |  |  |  |  |  |
|         | NCRFA0           |              |              |            | NCTFA      | A0[7:0]    |            |           |           |  |  |  |  |  |
|         | NCNTFA1          |              |              |            | NCNTF      | A1[7:0]    |            |           |           |  |  |  |  |  |
|         | NCRFA1           |              |              |            | NCTFA      | \1[7:0]    |            |           |           |  |  |  |  |  |
|         | NCNTFA2          |              |              |            | NCNTF      | A2[7:0]    |            |           |           |  |  |  |  |  |
|         | NCRFA2           |              |              |            | NCTFA      | A2[7:0]    |            |           |           |  |  |  |  |  |
|         | NCNTFA3          |              |              |            | NCNTF      | A3[7:0]    |            |           |           |  |  |  |  |  |
|         | NCRFA3           |              | NCTFA3[7:0]  |            |            |            |            |           |           |  |  |  |  |  |
|         | NCNTFA4          |              | NCNTFA4[7:0] |            |            |            |            |           |           |  |  |  |  |  |
|         | NCRFA4           |              | NCTFA4[7:0]  |            |            |            |            |           |           |  |  |  |  |  |
|         | NCNTFA5          |              |              |            | NCNTF      | A5[7:0]    |            |           |           |  |  |  |  |  |
|         | NCRFA5           |              |              |            | NCTFA      | A5[7:0]    |            |           |           |  |  |  |  |  |
|         | NCNTFA6          |              |              |            | NCNTF      | A6[7:0]    |            |           |           |  |  |  |  |  |
|         | NCRFA6           |              | NCTFA6[7:0]  |            |            |            |            |           |           |  |  |  |  |  |
|         | NCNTFA7          |              | NCNTFA7[7:0] |            |            |            |            |           |           |  |  |  |  |  |
|         | NCRFA7           |              | NCTFA7[7:0]  |            |            |            |            |           |           |  |  |  |  |  |
|         | NCNTFA8          |              | NCNTFA8[7:0] |            |            |            |            |           |           |  |  |  |  |  |
|         | NCRFA8           |              |              |            | NCTFA      | A8[7:0]    |            |           |           |  |  |  |  |  |
|         | NCNTFA9          |              |              |            | NCNTF      | A9[7:0]    |            |           |           |  |  |  |  |  |
|         | NCRFA9           |              |              |            | NCTFA      | A9[7:0]    |            |           |           |  |  |  |  |  |
|         | NCNTFA10         |              |              |            | NCNTFA     | A10[7:0]   |            |           |           |  |  |  |  |  |
|         | NCRFA10          |              |              |            | NCTFA      | 10[7:0]    |            |           |           |  |  |  |  |  |
|         | NCNTFA11         |              |              |            | NCNTFA     | A11[7:0]   |            |           |           |  |  |  |  |  |
|         | NCRFA11          |              |              |            | NCTFA      | 11[7:0]    |            |           |           |  |  |  |  |  |
|         | NCNTFA12         |              |              |            | NCNTFA     | A12[7:0]   |            |           |           |  |  |  |  |  |
|         | NCRFA12          |              |              |            | NCTFA      | 12[7:0]    |            |           |           |  |  |  |  |  |
|         | NCNTFA13         |              |              |            | NCNTFA     | A13[7:0]   |            |           |           |  |  |  |  |  |
|         | NCRFA13          | NCTFA13[7:0] |              |            |            |            |            |           |           |  |  |  |  |  |
|         | NCNTFA14         |              |              |            | NCNTFA     | A14[7:0]   |            |           |           |  |  |  |  |  |
| 1       | NCRFA14          |              |              |            | NCTFA      | 14[7:0]    |            |           |           |  |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3       | 26/18/10/2 | 25/17/9/1    | 24/16/8/0 |  |  |  |  |
|---------|------------------|--------------|--------------|------------|------------|------------------|------------|--------------|-----------|--|--|--|--|
| ATU-III | NCNTFA15         |              |              | 1          | NCNTF      | A15[7:0]         |            | L            | 1         |  |  |  |  |
|         | NCRFA15          |              |              |            | NCTFA      | 15[7:0]          |            |              |           |  |  |  |  |
|         | NCNTFA16         |              |              |            | NCNTF      | A16[7:0]         |            |              |           |  |  |  |  |
|         | NCRFA16          |              |              |            | NCTFA      | 16[7:0]          |            |              |           |  |  |  |  |
|         | NCNTFA17         |              |              |            | NCNTF      | <b>A</b> 17[7:0] |            |              |           |  |  |  |  |
|         | NCRFA17          |              | NCTFA17[7:0] |            |            |                  |            |              |           |  |  |  |  |
|         | NCNTFA18         |              |              |            | NCNTF      | A18[7:0]         |            |              |           |  |  |  |  |
|         | NCRFA18          |              |              |            | NCTFA      | 18[7:0]          |            |              |           |  |  |  |  |
|         | NCNTFA19         |              |              |            | NCNTF      | A19[7:0]         |            |              |           |  |  |  |  |
|         | NCRFA19          |              |              |            | NCTFA      | 19[7:0]          |            |              |           |  |  |  |  |
|         | NCNTFB0          |              |              |            | NCNTF      | B0[7:0]          |            |              |           |  |  |  |  |
|         | NCRFB0           |              | NCTFB0[7:0]  |            |            |                  |            |              |           |  |  |  |  |
|         | NCNTFB1          |              | NCNTFB1[7:0] |            |            |                  |            |              |           |  |  |  |  |
|         | NCRFB1           |              | NCTFB1[7:0]  |            |            |                  |            |              |           |  |  |  |  |
|         | NCNTFB2          | NCNTFB2[7:0] |              |            |            |                  |            |              |           |  |  |  |  |
|         | NCRFB2           | NCTFB2[7:0]  |              |            |            |                  |            |              |           |  |  |  |  |
|         | TCRF0            | C            | CKSELF0[2:0  | D]         |            | MDF0[2:0]        |            | EGSELF0[1:0] |           |  |  |  |  |
|         | TIERF0           | _            |              | _          | _          | OVECF0           | OVEBF0     | OVEAF0       | ICEF0     |  |  |  |  |
|         | TSRF0            | _            | _            | _          | _          | OVFCF0           | OVFBF0     | OVFAF0       | ICFF0     |  |  |  |  |
|         | ECNTAF0          |              |              |            | ECNTAF     | 0[23:16]         | •          |              |           |  |  |  |  |
|         |                  |              |              |            | ECNTAI     | F0[15:8]         |            |              |           |  |  |  |  |
|         |                  |              |              |            | ECNTA      | F0[7:0]          |            |              |           |  |  |  |  |
|         |                  | _            | _            | _          | _          | _                | _          | _            | _         |  |  |  |  |
|         | ECNTBF0          |              |              |            | ECNTBI     | F0[15:8]         |            |              | •         |  |  |  |  |
|         |                  |              |              |            | ECNTB      | F0[7:0]          |            |              |           |  |  |  |  |
|         | GRBF0            | GRBF0[15:8]  |              |            |            |                  |            |              |           |  |  |  |  |
|         |                  | GRBF0[7:0]   |              |            |            |                  |            |              |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7  | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|-------------|--------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | ECNTCF0          |             | ı            | <u>I</u>   | ECNTCF     | 0[23:16]   | <u>I</u>   | l .       | I         |  |  |  |  |
|         |                  |             |              |            | ECNTC      | F0[15:8]   |            |           |           |  |  |  |  |
|         |                  |             |              |            | ECNTC      | F0[7:0]    |            |           |           |  |  |  |  |
|         |                  |             | _            | _          |            |            |            | _         | _         |  |  |  |  |
|         | GRAF0            |             | GRAF0[23:16] |            |            |            |            |           |           |  |  |  |  |
|         |                  | GRAF0[15:8] |              |            |            |            |            |           |           |  |  |  |  |
|         |                  |             |              |            | GRAF       | 0[7:0]     |            |           |           |  |  |  |  |
|         |                  |             | _            |            | _          | _          |            |           | _         |  |  |  |  |
|         | CDRF0            |             |              |            | CDRF0      | [23:16]    |            |           |           |  |  |  |  |
|         |                  |             | CDRF0[15:8]  |            |            |            |            |           |           |  |  |  |  |
|         |                  |             |              |            | CDRF       | 0[7:0]     |            |           |           |  |  |  |  |
|         |                  | _           | _            | _          | _          | _          | _          | _         | _         |  |  |  |  |
|         | GRCF0            |             |              |            | GRCF0      | [23:16]    |            |           |           |  |  |  |  |
|         |                  |             | GRCF0[15:8]  |            |            |            |            |           |           |  |  |  |  |
|         |                  |             |              |            | GRCF       | 0[7:0]     |            |           |           |  |  |  |  |
|         |                  | _           |              | _          | _          |            |            | _         | _         |  |  |  |  |
|         | TCRF1            | (           | CKSELF1[2:0  | )]         |            | MDF1[2:0]  |            | EGSEI     | _F1[1:0]  |  |  |  |  |
|         | TIERF1           | _           | _            |            | _          | OVECF1     | OVEBF1     | OVEAF1    | ICEF1     |  |  |  |  |
|         | TSRF1            | _           |              | _          |            | OVFCF1     | OVFBF1     | OVFAF1    | ICFF1     |  |  |  |  |
|         | ECNTAF1          |             |              |            | ECNTAF     | 1[23:16]   |            |           |           |  |  |  |  |
|         |                  |             |              |            | ECNTAI     | F1[15:8]   |            |           |           |  |  |  |  |
|         |                  |             |              |            | ECNTA      | F1[7:0]    |            |           |           |  |  |  |  |
|         |                  | _           |              |            | _          | _          |            |           | _         |  |  |  |  |
|         | ECNTBF1          |             |              |            | ECNTBI     | F1[15:8]   |            |           |           |  |  |  |  |
|         |                  |             | ECNTBF1[7:0] |            |            |            |            |           |           |  |  |  |  |
|         | GRBF1            | GRBF1[15:8] |              |            |            |            |            |           |           |  |  |  |  |
|         |                  |             | GRBF1[7:0]   |            |            |            |            |           |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6     | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |
|---------|------------------|--------------|----------------|------------|------------|------------|------------|-----------|-----------|--|--|--|
| ATU-III | ECNTCF1          |              | I.             | <u>I</u>   | ECNTCF     | 1[23:16]   | l          |           |           |  |  |  |
|         |                  |              |                |            | ECNTC      | =1[15:8]   |            |           |           |  |  |  |
|         |                  |              |                |            | ECNTC      | F1[7:0]    |            |           |           |  |  |  |
|         |                  |              | _              | _          | _          | _          | _          | _         | _         |  |  |  |
|         | GRAF1            | GRAF1[23:16] |                |            |            |            |            |           |           |  |  |  |
|         |                  | GRAF1[15:8]  |                |            |            |            |            |           |           |  |  |  |
|         |                  |              | GRAF1[7:0]     |            |            |            |            |           |           |  |  |  |
|         |                  |              | _              | _          | _          | _          | _          |           |           |  |  |  |
|         | CDRF1            |              |                |            | CDRF1      | [23:16]    |            |           |           |  |  |  |
|         |                  |              | CDRF1[15:8]    |            |            |            |            |           |           |  |  |  |
|         |                  |              |                |            | CDRF       | 1[7:0]     |            |           |           |  |  |  |
|         |                  | _            |                | _          | _          |            | _          | _         | _         |  |  |  |
|         | GRCF1            |              | GRCF1[23:16]   |            |            |            |            |           |           |  |  |  |
|         |                  |              | GRCF1[15:8]    |            |            |            |            |           |           |  |  |  |
|         |                  |              | GRCF1[7:0]     |            |            |            |            |           |           |  |  |  |
|         |                  | _            | _              | _          | _          |            | _          | —         |           |  |  |  |
|         | TCRF2            | C            | CKSELF2[2:0    | 0]         |            | MDF2[2:0]  |            | EGSE      | LF2[1:0]  |  |  |  |
|         | TIERF2           | _            |                | _          | _          | OVECF2     | OVEBF2     | OVEAF2    | ICEF2     |  |  |  |
|         | TSRF2            |              |                | _          | _          | OVFCF2     | OVFBF2     | OVFAF2    | ICFF2     |  |  |  |
|         | ECNTAF2          |              | ECNTAF2[23:16] |            |            |            |            |           |           |  |  |  |
|         |                  |              |                |            | ECNTA      | F2[15:8]   |            |           |           |  |  |  |
|         |                  |              |                |            | ECNTA      | F2[7:0]    |            |           |           |  |  |  |
|         |                  | _            | _              | _          | _          | _          | _          | _         |           |  |  |  |
|         | ECNTBF2          |              |                |            | ECNTBI     | F2[15:8]   |            |           |           |  |  |  |
|         |                  |              |                |            | ECNTB      | F2[7:0]    |            |           |           |  |  |  |
|         | GRBF2            |              |                |            | GRBF2      | 2[15:8]    |            |           |           |  |  |  |
|         |                  | GRBF2[7:0]   |                |            |            |            |            |           |           |  |  |  |
|         | ECNTCF2          |              |                |            | ECNTCF     | 2[23:16]   |            |           |           |  |  |  |
|         |                  |              |                |            | ECNTC      | F2[15:8]   |            |           |           |  |  |  |
|         |                  |              |                |            | ECNTC      | F2[7:0]    |            |           |           |  |  |  |
|         |                  |              |                |            |            |            |            |           |           |  |  |  |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6               | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |
|---------|------------------|--------------|--------------------------|------------|------------|------------|------------|-----------|-----------|--|--|--|
| ATU-III | GRAF2            |              |                          | l          | GRAF2      | [23:16]    | I          | ļ.        | · I       |  |  |  |
|         |                  |              |                          |            | GRAF       | 2[15:8]    |            |           |           |  |  |  |
|         |                  |              |                          |            | GRAF       | 2[7:0]     |            |           |           |  |  |  |
|         |                  | _            | _                        | _          | _          | _          | _          | _         | _         |  |  |  |
|         | CDRF2            | CDRF2[23:16] |                          |            |            |            |            |           |           |  |  |  |
|         |                  | CDRF2[15:8]  |                          |            |            |            |            |           |           |  |  |  |
|         |                  |              | CDRF2[7:0]               |            |            |            |            |           |           |  |  |  |
|         |                  | _            | _                        | _          | _          | _          | _          | _         | _         |  |  |  |
|         | GRCF2            |              |                          |            | GRCF2      | [23:16]    |            |           |           |  |  |  |
|         |                  |              | GRCF2[15:8]              |            |            |            |            |           |           |  |  |  |
|         |                  |              |                          |            | GRCF       | 2[7:0]     |            |           |           |  |  |  |
|         |                  |              | _                        | _          |            | _          | _          |           | _         |  |  |  |
|         | TCRF3            | C            | CKSELF3[2:0] MDF3[2:0] E |            |            |            |            |           |           |  |  |  |
|         | TIERF3           |              | _                        | _          |            | OVECF3     | OVEBF3     | OVEAF3    | ICEF3     |  |  |  |
|         | TSRF3            |              | _                        | _          | _          | OVFCF3     | OVFBF3     | OVFAF3    | ICFF3     |  |  |  |
|         | ECNTAF3          |              | ECNTAF3[23:16]           |            |            |            |            |           |           |  |  |  |
|         |                  |              | ECNTAF3[15:8]            |            |            |            |            |           |           |  |  |  |
|         |                  |              |                          |            | ECNTA      | F3[7:0]    |            |           |           |  |  |  |
|         |                  | _            | _                        | _          | _          | _          | _          |           | _         |  |  |  |
|         | ECNTBF3          |              |                          |            | ECNTBI     | F3[15:8]   |            |           |           |  |  |  |
|         |                  |              |                          |            | ECNTB      | F3[7:0]    |            |           |           |  |  |  |
|         | GRBF3            |              |                          |            | GRBF       | 3[15:8]    |            |           |           |  |  |  |
|         |                  |              |                          |            | GRBF       | 3[7:0]     |            |           |           |  |  |  |
|         | ECNTCF3          |              |                          |            | ECNTCF     | 3[23:16]   |            |           |           |  |  |  |
|         |                  |              |                          |            | ECNTC      | F3[15:8]   |            |           |           |  |  |  |
|         |                  |              |                          |            | ECNTC      | F3[7:0]    |            |           |           |  |  |  |
|         |                  | _            | _                        | _          | _          | _          | _          | _         | _         |  |  |  |
|         | GRAF3            |              | GRAF3[23:16]             |            |            |            |            |           |           |  |  |  |
|         |                  |              | GRAF3[15:8]              |            |            |            |            |           |           |  |  |  |
|         |                  |              |                          |            | GRAF       | 3[7:0]     | _          | _         |           |  |  |  |
|         |                  |              | _                        | _          |            | _          | _          | _         |           |  |  |  |

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6                    | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |
|---------|------------------|---------------|-------------------------------|------------|------------|------------|------------|-----------|-----------|--|--|--|
| ATU-III | CDRF3            |               | <u>I</u>                      | <u>I</u>   | CDRF3      | [23:16]    | <u>I</u>   |           | I.        |  |  |  |
|         |                  |               |                               |            | CDRF       | 3[15:8]    |            |           |           |  |  |  |
|         |                  |               |                               |            | CDRF       | 3[7:0]     |            |           |           |  |  |  |
|         |                  | _             | _                             | _          | _          |            | _          | _         | _         |  |  |  |
|         | GRCF3            |               | GRCF3[23:16]                  |            |            |            |            |           |           |  |  |  |
|         |                  |               |                               |            | GRCF       | 3[15:8]    |            |           |           |  |  |  |
|         |                  |               | GRCF3[7:0]                    |            |            |            |            |           |           |  |  |  |
|         |                  |               |                               | _          | _          |            | _          |           | _         |  |  |  |
|         | TCRF4            | (             | CKSELF4[2:0] MDF4[2:0] EGSELF |            |            |            |            |           |           |  |  |  |
|         | TIERF4           | _             | _                             | _          | _          | OVECF4     | OVEBF4     | OVEAF4    | ICEF4     |  |  |  |
|         | TSRF4            | _             | U OVFCF4 OVFBF4 OVFAF4 I      |            |            |            |            |           |           |  |  |  |
|         | ECNTAF4          |               | ECNTAF4[23:16]                |            |            |            |            |           |           |  |  |  |
|         |                  |               | ECNTAF4[15:8]                 |            |            |            |            |           |           |  |  |  |
|         |                  |               | 1                             | 1          | ECNTA      | F4[7:0]    | 1          | T         | ı         |  |  |  |
|         |                  | _             |                               | _          |            |            | _          |           | _         |  |  |  |
|         | ECNTBF4          | ECNTBF4[15:8] |                               |            |            |            |            |           |           |  |  |  |
|         |                  |               | ECNTBF4[7:0]                  |            |            |            |            |           |           |  |  |  |
|         | GRBF4            |               |                               |            | GRBF4      |            |            |           |           |  |  |  |
|         |                  |               |                               |            | GRBF       |            |            |           |           |  |  |  |
|         | ECNTCF4          |               |                               |            | ECNTCF     |            |            |           |           |  |  |  |
|         |                  |               |                               |            | ECNTCI     |            |            |           |           |  |  |  |
|         |                  |               | I                             | I          | ECNTC      | F4[7:0]    | I          | 1         | 1         |  |  |  |
|         |                  |               | _                             | _          | _          | _          | _          | _         |           |  |  |  |
|         | GRAF4            |               |                               |            | GRAF4      |            |            |           |           |  |  |  |
|         |                  |               |                               |            | GRAF4      |            |            |           |           |  |  |  |
|         |                  |               | ı                             | ı          | GRAF       | 4[7:0]     | ı          | I         |           |  |  |  |
| i       |                  |               |                               |            |            |            |            |           |           |  |  |  |

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6     | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |
|---------|------------------|---------------|----------------|------------|------------|------------|------------|-----------|-----------|--|--|--|
| ATU-III | CDRF4            |               |                |            | CDRF4      | [23:16]    | •          | Į.        | L         |  |  |  |
|         |                  |               |                |            | CDRF4      | 4[15:8]    |            |           |           |  |  |  |
|         |                  |               |                |            | CDRF       | 4[7:0]     |            |           |           |  |  |  |
|         |                  | _             |                |            | _          |            |            |           | _         |  |  |  |
|         | GRCF4            | GRCF4[23:16]  |                |            |            |            |            |           |           |  |  |  |
|         |                  |               | GRCF4[15:8]    |            |            |            |            |           |           |  |  |  |
|         |                  |               |                |            | GRCF       | 4[7:0]     |            |           |           |  |  |  |
|         |                  |               | _              |            |            |            |            | _         | _         |  |  |  |
|         | TCRF5            | (             | CKSELF5[2:0    | 0]         |            | MDF5[2:0]  |            | EGSEL     | F5[1:0]   |  |  |  |
|         | TIERF5           | _             | _              | _          |            | OVECF5     | OVEBF5     | OVEAF5    | ICEF5     |  |  |  |
|         | TSRF5            | _             |                |            |            | OVFCF5     | OVFBF5     | OVFAF5    | ICFF5     |  |  |  |
|         | ECNTAF5          |               | ECNTAF5[23:16] |            |            |            |            |           |           |  |  |  |
|         |                  |               | ECNTAF5[15:8]  |            |            |            |            |           |           |  |  |  |
|         |                  |               |                |            | ECNTA      | F5[7:0]    |            |           |           |  |  |  |
|         |                  |               | _              |            |            |            |            |           | _         |  |  |  |
|         | ECNTBF5          | ECNTBF5[15:8] |                |            |            |            |            |           |           |  |  |  |
|         |                  |               |                |            | ECNTB      | F5[7:0]    |            |           |           |  |  |  |
|         | GRBF5            |               |                |            | GRBF       | 5[15:8]    |            |           |           |  |  |  |
|         |                  |               |                |            | GRBF       | 5[7:0]     |            |           |           |  |  |  |
|         | ECNTCF5          |               |                |            | ECNTCF     | 5[23:16]   |            |           |           |  |  |  |
|         |                  |               |                |            | ECNTCI     | F5[15:8]   |            |           |           |  |  |  |
|         |                  |               | 1              | 1          | ECNTC      | F5[7:0]    | 1          | 1         | 1         |  |  |  |
|         |                  | _             |                | _          | _          |            |            |           | _         |  |  |  |
|         | GRAF5            |               |                |            | GRAF5      | [23:16]    |            |           |           |  |  |  |
|         |                  |               |                |            | GRAF       | 5[15:8]    |            |           |           |  |  |  |
|         |                  |               | GRAF5[7:0]     |            |            |            |            |           |           |  |  |  |
|         |                  |               | _              | _          | _          |            |            | _         | _         |  |  |  |

| Module  | Register<br>Name | 31/23/15/7     | 30/22/14/6                  | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |
|---------|------------------|----------------|-----------------------------|------------|------------|------------|------------|-----------|-----------|--|--|--|
| ATU-III | CDRF5            |                |                             |            | CDRF5      | [23:16]    |            |           |           |  |  |  |
|         |                  |                |                             |            | CDRF       | 5[15:8]    |            |           |           |  |  |  |
|         |                  |                |                             |            | CDRF       | 5[7:0]     |            |           |           |  |  |  |
|         |                  |                |                             |            |            |            |            | _         |           |  |  |  |
|         | GRCF5            |                |                             |            | GRCF5      | [23:16]    |            |           |           |  |  |  |
|         |                  |                |                             |            | GRCF       | 5[15:8]    |            |           |           |  |  |  |
|         |                  |                |                             |            | GRCF       | 5[7:0]     |            |           |           |  |  |  |
|         |                  | _              |                             |            | _          |            |            | _         |           |  |  |  |
|         | TCRF6            | (              | CKSELF6[2:0] MDF6[2:0] EGSE |            |            |            |            |           |           |  |  |  |
|         | TIERF6           | _              | _                           |            | _          | OVECF6     | OVEBF6     | OVEAF6    | ICEF6     |  |  |  |
|         | TSRF6            | _              | OVFCF6 OVFBF6 OVFAF6        |            |            |            |            |           |           |  |  |  |
|         | ECNTAF6          |                | ECNTAF6[23:16]              |            |            |            |            |           |           |  |  |  |
|         |                  |                | ECNTAF6[15:8]               |            |            |            |            |           |           |  |  |  |
|         |                  |                |                             |            | ECNTA      | F6[7:0]    |            |           |           |  |  |  |
|         |                  | _              | _                           | _          | _          | _          | _          |           |           |  |  |  |
|         | ECNTBF6          | ECNTBF6[15:8]  |                             |            |            |            |            |           |           |  |  |  |
|         |                  |                | ECNTBF6[7:0]                |            |            |            |            |           |           |  |  |  |
|         | GRBF6            | GRBF6[15:8]    |                             |            |            |            |            |           |           |  |  |  |
|         |                  |                | GRBF6[7:0]                  |            |            |            |            |           |           |  |  |  |
|         | ECNTCF6          | ECNTCF6[23:16] |                             |            |            |            |            |           |           |  |  |  |
|         |                  |                |                             |            | ECNTC      | F6[15:8]   |            |           |           |  |  |  |
|         |                  |                | T                           | T          | ECNTC      | F6[7:0]    | T          | T         | 1         |  |  |  |
|         |                  | _              |                             | _          | _          |            | _          |           |           |  |  |  |
|         | GRAF6            |                |                             |            | GRAF6      | [23:16]    |            |           |           |  |  |  |
|         |                  |                |                             |            | GRAF       | 6[15:8]    |            |           |           |  |  |  |
|         |                  |                | ı                           | ı          | GRAF       | 6[7:0]     | ı          | 1         | 1         |  |  |  |
|         |                  | _              | _                           | _          | _          | _          | _          | _         |           |  |  |  |
|         | CDRF6            |                |                             |            | CDRF6      |            |            |           |           |  |  |  |
|         |                  |                | CDRF6[15:8]                 |            |            |            |            |           |           |  |  |  |
|         |                  |                | CDRF6[7:0]                  |            |            |            |            |           |           |  |  |  |
|         |                  | _              | _                           | _          | _          | _          | _          | _         | _         |  |  |  |

| Module  | Register<br>Name | 31/23/15/7     | 30/22/14/6    | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |  |
|---------|------------------|----------------|---------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|--|
| ATU-III | GRCF6            |                | I .           |            | GRCF6      | [23:16]    |            | l         | ı         |  |  |  |  |  |
|         |                  |                |               |            | GRCF       | 6[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |                |               |            | GRCF       | 6[7:0]     |            |           |           |  |  |  |  |  |
|         |                  |                |               | _          | _          | _          |            | _         | _         |  |  |  |  |  |
|         | TCRF7            | (              | CKSELF7[2:0   | 0]         |            | MDF7[2:0]  |            | EGSEL     | F7[1:0]   |  |  |  |  |  |
|         | TIERF7           | _              | _             | _          | _          | OVECF7     | OVEBF7     | OVEAF7    | ICEF7     |  |  |  |  |  |
|         | TSRF7            | _              | _             | _          | _          | OVFCF7     | OVFBF7     | OVFAF7    | ICFF7     |  |  |  |  |  |
|         | ECNTAF7          | ECNTAF7[23:16] |               |            |            |            |            |           |           |  |  |  |  |  |
|         |                  |                | ECNTAF7[15:8] |            |            |            |            |           |           |  |  |  |  |  |
|         |                  |                |               |            | ECNTA      | .F7[7:0]   |            |           |           |  |  |  |  |  |
|         |                  | _              | _             | _          |            |            | _          |           | _         |  |  |  |  |  |
|         | ECNTBF7          |                | ECNTBF7[15:8] |            |            |            |            |           |           |  |  |  |  |  |
|         |                  |                |               |            | ECNTB      | F7[7:0]    |            |           |           |  |  |  |  |  |
|         | GRBF7            | GRBF7[15:8]    |               |            |            |            |            |           |           |  |  |  |  |  |
|         |                  |                |               |            |            |            |            |           |           |  |  |  |  |  |
|         | ECNTCF7          |                |               |            |            |            |            |           |           |  |  |  |  |  |
|         |                  | ECNTCF7[15:8]  |               |            |            |            |            |           |           |  |  |  |  |  |
|         |                  |                |               |            | ECNTC      | F7[7:0]    |            |           |           |  |  |  |  |  |
|         |                  | _              | _             |            |            |            |            |           | _         |  |  |  |  |  |
|         | GRAF7            |                | GRAF7[23:16]  |            |            |            |            |           |           |  |  |  |  |  |
|         |                  |                |               |            | GRAF       | 7[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |                |               |            | GRAF       | 7[7:0]     |            |           |           |  |  |  |  |  |
|         |                  | _              | _             |            |            | _          | _          |           | _         |  |  |  |  |  |
|         | CDRF7            |                |               |            | CDRF7      | [23:16]    |            |           |           |  |  |  |  |  |
|         |                  |                |               |            | CDRF       | 7[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |                |               |            | CDRF       | 7[7:0]     |            |           |           |  |  |  |  |  |
|         |                  |                | _             | _          | _          | _          | _          |           | _         |  |  |  |  |  |
|         | GRCF7            |                |               |            | GRCF7      | [23:16]    |            |           |           |  |  |  |  |  |
|         |                  | GRCF7[15:8]    |               |            |            |            |            |           |           |  |  |  |  |  |
|         |                  |                |               |            | GRCF       | 7[7:0]     |            |           |           |  |  |  |  |  |
|         |                  |                |               |            |            |            |            |           |           |  |  |  |  |  |

Section 31 List of Registers

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6    | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|--------------|---------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | TSRF9            | _            |               |            |            | OVFCF9     | OVFBF9     | OVFAF9    | ICFF9     |  |  |  |  |
|         | ECNTAF9          |              |               |            | ECNTAF     | 9[23:16]   |            |           |           |  |  |  |  |
|         |                  |              | ECNTAF9[15:8] |            |            |            |            |           |           |  |  |  |  |
|         |                  |              |               |            | ECNTA      | F9[7:0]    |            |           |           |  |  |  |  |
|         |                  | _            |               |            |            |            |            |           |           |  |  |  |  |
|         | ECNTBF9          |              |               |            | ECNTB      | F9[15:8]   |            |           |           |  |  |  |  |
|         |                  |              |               |            | ECNTB      | F9[7:0]    |            |           |           |  |  |  |  |
|         | GRBF9            |              |               |            | GRBF       | 9[15:8]    |            |           |           |  |  |  |  |
|         |                  |              |               |            | GRBF       | 9[7:0]     |            |           |           |  |  |  |  |
|         | ECNTCF9          |              |               |            | ECNTCF     | 9[23:16]   |            |           |           |  |  |  |  |
|         |                  |              |               |            | ECNTC      | -9[15:8]   |            |           |           |  |  |  |  |
|         |                  |              |               |            | ECNTC      | F9[7:0]    |            |           |           |  |  |  |  |
|         |                  | _            |               |            |            |            |            |           |           |  |  |  |  |
|         | GRAF9            | GRAF9[23:16] |               |            |            |            |            |           |           |  |  |  |  |
|         |                  | GRAF9[15:8]  |               |            |            |            |            |           |           |  |  |  |  |
|         |                  |              | 1             | 1          | GRAF       | 9[7:0]     | 1          | 1         |           |  |  |  |  |
|         |                  | _            |               | _          |            | _          |            |           | _         |  |  |  |  |
|         | CDRF9            | CDRF9[23:16] |               |            |            |            |            |           |           |  |  |  |  |
|         |                  | CDRF9[15:8]  |               |            |            |            |            |           |           |  |  |  |  |
|         |                  |              | 1             | 1          | CDRF       | 9[7:0]     | 1          | 1         |           |  |  |  |  |
|         |                  | _            |               | _          |            | _          |            |           | _         |  |  |  |  |
|         | GRCF9            |              |               |            | GRCF9      | [23:16]    |            |           |           |  |  |  |  |
|         |                  |              |               |            | GRCF       | 9[15:8]    |            |           |           |  |  |  |  |
|         |                  |              | 1             | 1          | GRCF       | 9[7:0]     | 1          | 1         |           |  |  |  |  |
|         |                  | _  -  -      |               |            |            |            |            |           |           |  |  |  |  |
|         | TCRF10           | С            | KSELF10[2:    | 0]         |            | MDF10[2:0] | 1          | EGSELI    | F10[1:0]  |  |  |  |  |
|         | TIERF10          |              | _             | _          | _          | OVECF10    | OVEBF10    | OVEAF10   | ICEF10    |  |  |  |  |
|         | TSRF10           |              |               |            |            | OVFCF10    | OVFBF10    | OVFAF10   | ICFF10    |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7                    | 30/22/14/6      | 29/21/13/5 | 28/20/12/4        | 27/19/11/3   | 26/18/10/2 | 25/17/9/1 | 24/16/8/0          |  |  |  |
|---------|------------------|-------------------------------|-----------------|------------|-------------------|--------------|------------|-----------|--------------------|--|--|--|
| ATU-III | ECNTAF10         |                               | •               |            | ECNTAF            | 10[23:16]    | •          | •         |                    |  |  |  |
|         |                  |                               |                 |            | ECNTAF            | 10[15:8]     |            |           |                    |  |  |  |
|         |                  |                               |                 |            | ECNTAI            | F10[7:0]     |            |           |                    |  |  |  |
|         |                  |                               |                 |            |                   |              |            |           |                    |  |  |  |
|         | ECNTBF10         |                               |                 |            | ECNTBF10[15:8]    |              |            |           |                    |  |  |  |
|         |                  |                               |                 |            | ECNTBI            | F10[7:0]     |            |           |                    |  |  |  |
|         | GRBF10           |                               |                 |            | GRBF1             | 0[15:8]      |            |           |                    |  |  |  |
|         |                  |                               | GRBF10[7:0]     |            |                   |              |            |           |                    |  |  |  |
|         | ECNTCF10         |                               | ECNTCF10[23:16] |            |                   |              |            |           |                    |  |  |  |
|         |                  |                               | ECNTCF10[15:8]  |            |                   |              |            |           |                    |  |  |  |
|         |                  |                               | 1               | F10[7:0]   | 1                 | 1            |            |           |                    |  |  |  |
|         |                  | _                             |                 | _          |                   |              | _          |           | _                  |  |  |  |
|         | GRAF10           | GRAF10[23:16]                 |                 |            |                   |              |            |           |                    |  |  |  |
|         |                  | GRAF10[15:8]                  |                 |            |                   |              |            |           |                    |  |  |  |
|         |                  |                               | ı               | ı          | GRAF              | 10[7:0]      | 1          | 1         |                    |  |  |  |
|         |                  | _                             |                 |            |                   |              | _          |           |                    |  |  |  |
|         | CDRF10           | CDRF10[23:16]                 |                 |            |                   |              |            |           |                    |  |  |  |
|         |                  | CDRF10[15:8]                  |                 |            |                   |              |            |           |                    |  |  |  |
|         |                  |                               | I               | I          | CDRF <sup>-</sup> | 10[7:0]<br>T | 1          | 1         | 1                  |  |  |  |
|         |                  | _                             |                 |            |                   | <u> </u>     |            |           |                    |  |  |  |
|         | GRCF10           | GRCF10[23:16]                 |                 |            |                   |              |            |           |                    |  |  |  |
|         |                  |                               |                 |            | GRCF1             |              |            |           |                    |  |  |  |
|         |                  |                               | 1               | 1          | GRCF              | 10[7:0]      |            |           |                    |  |  |  |
|         | TODE11           |                               |                 | 01         |                   | MDE11[0:0]   |            |           | F14[4:0]           |  |  |  |
|         | TCRF11           |                               | KSELF11[2:      | 0]         |                   | MDF11[2:0]   | OVEBF11    | OVEAF11   | F11[1:0]<br>ICEF11 |  |  |  |
|         | TSRF11           |                               |                 |            |                   | OVECF11      | OVEBF11    | OVEAF11   | ICFF11             |  |  |  |
|         | ECNTAF11         |                               |                 |            | ECNTAF            | L            | OVIBITI    | OVIAITI   |                    |  |  |  |
|         | LONTAITI         |                               |                 |            |                   |              |            |           |                    |  |  |  |
|         |                  | ECNTAF11[15:8]  ECNTAF11[7:0] |                 |            |                   |              |            |           |                    |  |  |  |
|         |                  |                               |                 |            |                   |              |            |           |                    |  |  |  |
|         |                  |                               |                 |            |                   | <u> </u>     |            |           |                    |  |  |  |

| Module  | Register<br>Name | 31/23/15/7      | 30/22/14/6    | 29/21/13/5 | 28/20/12/4        | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|-----------------|---------------|------------|-------------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | ECNTBF11         |                 |               |            | ECNTBF            | 11[15:8]   |            | •         | <u> </u>  |  |  |  |  |
|         |                  |                 |               |            | ECNTBI            | F11[7:0]   |            |           |           |  |  |  |  |
|         | GRBF11           |                 |               |            | GRBF1             | 1[15:8]    |            |           |           |  |  |  |  |
|         |                  |                 |               |            | GRBF <sup>-</sup> | 11[7:0]    |            |           |           |  |  |  |  |
|         | ECNTCF11         | ECNTCF11[23:16] |               |            |                   |            |            |           |           |  |  |  |  |
|         |                  | ECNTCF11[15:8]  |               |            |                   |            |            |           |           |  |  |  |  |
|         |                  |                 | ECNTCF11[7:0] |            |                   |            |            |           |           |  |  |  |  |
|         |                  | _               | _             |            | _                 | _          |            |           | _         |  |  |  |  |
|         | GRAF11           | GRAF11[23:16]   |               |            |                   |            |            |           |           |  |  |  |  |
|         |                  |                 | GRAF11[15:8]  |            |                   |            |            |           |           |  |  |  |  |
|         |                  |                 | GRAF11[7:0]   |            |                   |            |            |           |           |  |  |  |  |
|         |                  | _               | _             |            | _                 |            |            | _         | _         |  |  |  |  |
|         | CDRF11           | CDRF11[23:16]   |               |            |                   |            |            |           |           |  |  |  |  |
|         |                  | CDRF11[15:8]    |               |            |                   |            |            |           |           |  |  |  |  |
|         |                  |                 | CDRF11[7:0]   |            |                   |            |            |           |           |  |  |  |  |
|         |                  | _               | _             | _          |                   |            | _          |           | _         |  |  |  |  |
|         | GRCF11           |                 |               |            | GRCF1             | 1[23:16]   |            |           |           |  |  |  |  |
|         |                  |                 |               |            | GRCF1             | 1[15:8]    |            |           |           |  |  |  |  |
|         |                  |                 | GRCF11[7:0]   |            |                   |            |            |           |           |  |  |  |  |
|         |                  | _               | _             |            | _                 |            |            | _         | _         |  |  |  |  |
|         | TCRF12           | С               | KSELF12[2:    | 0]         |                   | MDF12[2:0] |            | EGSEL     | F12[1:0]  |  |  |  |  |
|         | TIERF12          | _               |               |            | _                 | OVECF12    | OVEBF12    | OVEAF12   | ICEF12    |  |  |  |  |
|         | TSRF12           | _               |               |            |                   | OVFCF12    | OVFBF12    | OVFAF12   | ICFF12    |  |  |  |  |
|         | ECNTAF12         |                 |               |            | ECNTAF            | 12[23:16]  |            |           |           |  |  |  |  |
|         |                  |                 |               |            | ECNTAF            | 12[15:8]   |            |           |           |  |  |  |  |
|         |                  |                 |               |            | ECNTAI            | F12[7:0]   |            |           |           |  |  |  |  |
|         |                  | _               | _             | _          | _                 |            | _          |           | _         |  |  |  |  |
|         | ECNTBF12         |                 |               |            | ECNTBF            | 12[15:8]   |            |           |           |  |  |  |  |
|         |                  |                 |               |            | ECNTBI            | F12[7:0]   |            |           |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6          | 29/21/13/5 | 28/20/12/4        | 27/19/11/3            | 26/18/10/2 | 25/17/9/1 | 24/16/8/0          |  |  |  |  |
|---------|------------------|---------------|---------------------|------------|-------------------|-----------------------|------------|-----------|--------------------|--|--|--|--|
| ATU-III | GRBF12           |               |                     | l          | GRBF1             | 2[15:8]               | l          | I         |                    |  |  |  |  |
|         |                  |               |                     |            | GRBF <sup>-</sup> | 12[7:0]               |            |           |                    |  |  |  |  |
|         | ECNTCF12         |               |                     |            | ECNTCF            | 12[23:16]             |            |           |                    |  |  |  |  |
|         |                  |               |                     |            | ECNTCF            | 12[15:8]              |            |           |                    |  |  |  |  |
|         |                  |               |                     |            | ECNTC             | F12[7:0]              |            |           |                    |  |  |  |  |
|         |                  | _             | _                   | _          | _                 | _                     | _          |           | _                  |  |  |  |  |
|         | GRAF12           |               |                     |            | GRAF12            | 2[23:16]              |            |           |                    |  |  |  |  |
|         |                  |               |                     |            | GRAF1             | 2[15:8]               |            |           |                    |  |  |  |  |
|         |                  |               |                     |            | GRAF              | 12[7:0]               |            |           | 1                  |  |  |  |  |
|         |                  | _             | -  -  -  -  -  -  - |            |                   |                       |            |           |                    |  |  |  |  |
|         | CDRF12           |               |                     |            | CDRF12            | 2[23:16]              |            |           |                    |  |  |  |  |
|         |                  |               | CDRF12[15:8]        |            |                   |                       |            |           |                    |  |  |  |  |
|         |                  |               | 1                   | 1          | CDRF-             | 12[7:0]               | 1          | 1         |                    |  |  |  |  |
|         |                  | _             |                     |            |                   |                       |            |           |                    |  |  |  |  |
|         | GRCF12           | GRCF12[23:16] |                     |            |                   |                       |            |           |                    |  |  |  |  |
|         |                  |               | GRCF12[15:8]        |            |                   |                       |            |           |                    |  |  |  |  |
|         |                  |               | 1                   | I          | GRCF <sup>-</sup> | 12[7:0]<br>T          | I          | 1         |                    |  |  |  |  |
|         |                  | _             |                     | _          | _                 |                       | _          |           | _                  |  |  |  |  |
|         | GRDF12           |               | GRDF12[23:16]       |            |                   |                       |            |           |                    |  |  |  |  |
|         |                  |               |                     |            | GRDF1             |                       |            |           |                    |  |  |  |  |
|         |                  |               | <u> </u>            | <u> </u>   | GRDF              | 12[7:0]<br>           | <u> </u>   | 1         |                    |  |  |  |  |
|         | TODE10           | _             |                     | -          | _                 |                       | _          |           |                    |  |  |  |  |
|         | TCRF13           |               | KSELF13[2:          | oj<br>I    |                   | MDF13[2:0]<br>OVECF13 | OVEBF13    | OVEAF13   | F13[1:0]<br>ICEF13 |  |  |  |  |
|         | TSRF13           | _             |                     |            |                   |                       |            |           |                    |  |  |  |  |
|         | ECNTAF13         |               |                     |            | ECNTAF            | OVFCF13               | OVFBF13    | OVFAF13   | ICFF13             |  |  |  |  |
|         | ECINTAFTS        |               |                     |            | ECNTAF            |                       |            |           |                    |  |  |  |  |
|         |                  |               |                     |            | ECNTAI            |                       |            |           |                    |  |  |  |  |
|         |                  |               |                     |            | LONTAI            |                       |            |           |                    |  |  |  |  |
|         |                  | _             |                     |            |                   |                       | _          | _         |                    |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7      | 30/22/14/6    | 29/21/13/5 | 28/20/12/4        | 27/19/11/3   | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|-----------------|---------------|------------|-------------------|--------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | ECNTBF13         |                 | •             | •          | ECNTBF            | 13[15:8]     | •          | •         | L         |  |  |  |  |
|         |                  |                 |               |            | ECNTB             | F13[7:0]     |            |           |           |  |  |  |  |
|         | GRBF13           |                 |               |            | GRBF1             | 3[15:8]      |            |           |           |  |  |  |  |
|         |                  |                 | GRBF13[7:0]   |            |                   |              |            |           |           |  |  |  |  |
|         | ECNTCF13         | ECNTCF13[23:16] |               |            |                   |              |            |           |           |  |  |  |  |
|         |                  |                 |               |            | ECNTCF            | 13[15:8]     |            |           |           |  |  |  |  |
|         |                  |                 | 1             | 1          | ECNTC             | F13[7:0]     | 1          | 1         |           |  |  |  |  |
|         |                  | _               |               |            |                   |              |            |           |           |  |  |  |  |
|         | GRAF13           |                 | GRAF13[23:16] |            |                   |              |            |           |           |  |  |  |  |
|         |                  |                 | GRAF13[15:8]  |            |                   |              |            |           |           |  |  |  |  |
|         |                  |                 | T             | T          | GRAF              | 13[7:0]      | T          | T         |           |  |  |  |  |
|         | CDRF13           | _               | _             |            |                   |              |            |           |           |  |  |  |  |
|         | CDRF13           | CDRF13[23:16]   |               |            |                   |              |            |           |           |  |  |  |  |
|         |                  |                 | CDRF13[15:8]  |            |                   |              |            |           |           |  |  |  |  |
|         |                  |                 | 1             | 1          | CDRF <sup>-</sup> | 13[7:0]<br>T | 1          | 1         |           |  |  |  |  |
|         |                  | _               |               |            | <u> </u>          | <u> </u>     |            |           |           |  |  |  |  |
|         | GRCF13           |                 |               |            | GRCF1             |              |            |           |           |  |  |  |  |
|         |                  |                 |               |            | GRCF1             |              |            |           |           |  |  |  |  |
|         |                  |                 | 1             | <u> </u>   | GRCF <sup>-</sup> | 13[7:0]<br>I | 1          | <u> </u>  |           |  |  |  |  |
|         | 0000540          |                 |               |            | - 000051          |              |            |           |           |  |  |  |  |
|         | GRDF13           |                 |               |            | GRDF1             |              |            |           |           |  |  |  |  |
|         |                  |                 |               |            | GRDF1             |              |            |           |           |  |  |  |  |
|         |                  |                 |               |            | GRDF <sup>-</sup> | 13[7:0]      |            |           |           |  |  |  |  |
|         | TCRF14           |                 | CKSELF14[2:0] |            |                   | MDF14[2:0]   |            | FGSFU     | F14[1:0]  |  |  |  |  |
|         | TIERF14          |                 |               | ~ <u>ı</u> |                   | OVECF14      | OVEBF14    | OVEAF14   | ICEF14    |  |  |  |  |
|         | TSRF14           |                 |               |            |                   | OVFCF14      | OVFBF14    | OVFAF14   | ICFF14    |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7                  | 30/22/14/6                    | 29/21/13/5 | 28/20/12/4          | 27/19/11/3   | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |
|---------|------------------|-----------------------------|-------------------------------|------------|---------------------|--------------|------------|-----------|-----------|--|--|--|
| ATU-III | ECNTAF14         |                             |                               |            | ECNTAF <sup>2</sup> | 14[23:16]    |            |           |           |  |  |  |
|         |                  |                             |                               |            | ECNTAF              | 14[15:8]     |            |           |           |  |  |  |
|         |                  |                             |                               |            | ECNTAI              | F14[7:0]     |            |           |           |  |  |  |
|         |                  | _                           |                               |            |                     |              |            |           | _         |  |  |  |
|         | ECNTBF14         |                             |                               |            | ECNTBF              | 14[15:8]     |            |           |           |  |  |  |
|         |                  |                             |                               |            | ECNTBI              | F14[7:0]     |            |           |           |  |  |  |
|         | GRBF14           |                             |                               |            | GRBF1               | 4[15:8]      |            |           |           |  |  |  |
|         |                  |                             |                               |            | GRBF <sup>-</sup>   | 14[7:0]      |            |           |           |  |  |  |
|         | ECNTCF14         |                             |                               |            | ECNTCF              | 14[23:16]    |            |           |           |  |  |  |
|         |                  |                             |                               |            | ECNTCF              | 14[15:8]     |            |           |           |  |  |  |
|         |                  |                             | ı                             | 1          | ECNTC               | F14[7:0]     | Π          | 1         | 1         |  |  |  |
|         |                  | _                           | GRAF14[23:16]                 |            |                     |              |            |           |           |  |  |  |
|         | GRAF14           |                             | GRAF14[23:16]<br>GRAF14[15:8] |            |                     |              |            |           |           |  |  |  |
|         |                  |                             | GRAF14[15:8]                  |            |                     |              |            |           |           |  |  |  |
|         |                  |                             | I                             |            | GRAF                | 14[7:0]<br>  |            |           |           |  |  |  |
|         | CDRF14           | _                           |                               |            |                     |              |            |           |           |  |  |  |
|         | CDRF14           | CDRF14[23:16]  CDRF14[15:8] |                               |            |                     |              |            |           |           |  |  |  |
|         |                  |                             | CDRF14[15:8]  CDRF14[7:0]     |            |                     |              |            |           |           |  |  |  |
|         |                  |                             |                               |            |                     |              |            |           |           |  |  |  |
|         | GRCF14           |                             |                               |            | GRCF1               | <br>4[23:16] |            |           |           |  |  |  |
|         |                  |                             |                               |            | GRCF1               |              |            |           |           |  |  |  |
|         |                  |                             |                               |            | GRCF                |              |            |           |           |  |  |  |
|         |                  | _                           | _                             | _          | _                   | _            | _          | _         |           |  |  |  |
|         | GRDF14           |                             | I                             |            | GRDF1               | 4[23:16]     |            |           | ı         |  |  |  |
|         |                  |                             |                               |            | GRDF1               | 4[15:8]      |            |           |           |  |  |  |
|         |                  |                             |                               |            | GRDF <sup>-</sup>   | 14[7:0]      |            |           |           |  |  |  |
|         |                  | _                           | _                             | _          | _                   | _            | _          | _         | _         |  |  |  |
|         | TCRF15           | С                           | KSELF15[2:                    | :0]        |                     | MDF15[2:0]   |            | EGSELI    | F15[1:0]  |  |  |  |
|         | TIERF15          | _                           | _                             | _          | _                   | OVECF15      | OVEBF15    | OVEAF15   | ICEF15    |  |  |  |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6      | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|--------------|-----------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | TSRF15           | _            | _               | _          | _          | OVFCF15    | OVFBF15    | OVFAF15   | ICFF15    |  |  |  |  |
|         | ECNTAF15         |              | •               |            | ECNTAF     | 15[23:16]  |            | •         | •         |  |  |  |  |
|         |                  |              |                 |            | ECNTAF     | 15[15:8]   |            |           |           |  |  |  |  |
|         |                  |              |                 |            | ECNTA      | F15[7:0]   |            |           |           |  |  |  |  |
|         |                  | _            |                 | _          | _          |            | _          |           | _         |  |  |  |  |
|         | ECNTBF15         |              | ECNTBF15[15:8]  |            |            |            |            |           |           |  |  |  |  |
|         |                  |              | ECNTBF15[7:0]   |            |            |            |            |           |           |  |  |  |  |
|         | GRBF15           |              | GRBF15[15:8]    |            |            |            |            |           |           |  |  |  |  |
|         |                  |              |                 |            | GRBF       | 15[7:0]    |            |           |           |  |  |  |  |
|         | ECNTCF15         |              | ECNTCF15[23:16] |            |            |            |            |           |           |  |  |  |  |
|         |                  |              |                 |            | ECNTCF     | 15[15:8]   |            |           |           |  |  |  |  |
|         |                  |              |                 |            | ECNTC      | F15[7:0]   |            |           |           |  |  |  |  |
|         |                  | _            | _               |            |            | _          |            | _         | _         |  |  |  |  |
|         | GRAF15           |              |                 |            | GRAF1      | 5[23:16]   |            |           |           |  |  |  |  |
|         |                  | GRAF15[15:8] |                 |            |            |            |            |           |           |  |  |  |  |
|         |                  |              |                 |            | GRAF       | 15[7:0]    |            |           |           |  |  |  |  |
|         |                  | _            |                 | _          | _          |            | _          |           | _         |  |  |  |  |
|         | CDRF15           |              |                 |            | CDRF1      | 5[23:16]   |            |           |           |  |  |  |  |
|         |                  |              |                 |            | CDRF1      | 5[15:8]    |            |           |           |  |  |  |  |
|         |                  |              |                 | 1          | CDRF       | 15[7:0]    | 1          |           |           |  |  |  |  |
|         |                  | _            | _               | _          |            | _          | _          |           | _         |  |  |  |  |
|         | GRCF15           |              |                 |            | GRCF1      | 5[23:16]   |            |           |           |  |  |  |  |
|         |                  |              | GRCF15[15:8]    |            |            |            |            |           |           |  |  |  |  |
|         |                  |              | 1               |            |            |            |            |           |           |  |  |  |  |
|         |                  | _            | _               | _          |            | _          | _          |           | _         |  |  |  |  |
|         | GRDF15           |              |                 |            | GRDF1      | 5[23:16]   |            |           |           |  |  |  |  |
|         |                  |              |                 |            | GRDF1      |            |            |           |           |  |  |  |  |
|         |                  |              | 1               | 1          | GRDF       | 15[7:0]    | 1          | 1         |           |  |  |  |  |
|         |                  |              |                 |            |            |            |            | <u> </u>  | <u>L</u>  |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6      | 29/21/13/5 | 28/20/12/4        | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |  |
|---------|------------------|---------------|-----------------|------------|-------------------|------------|------------|-----------|-----------|--|--|--|--|--|
| ATU-III | TCRF16           | С             | KSELF16[2:      | 0]         |                   | MDF16[2:0] | <u>I</u>   | EGSELF    | F16[1:0]  |  |  |  |  |  |
|         | TIERF16          |               |                 |            |                   | OVECF16    | OVEBF16    | OVEAF16   | ICEF16    |  |  |  |  |  |
|         | TSRF16           | _             | _               | _          | _                 | OVFCF16    | OVFBF16    | OVFAF16   | ICFF16    |  |  |  |  |  |
|         | ECNTAF16         |               |                 |            | ECNTAF.           | 16[23:16]  |            |           | ı         |  |  |  |  |  |
|         |                  |               |                 |            | ECNTAF            | 16[15:8]   |            |           |           |  |  |  |  |  |
|         |                  |               |                 |            | ECNTAF            | 16[7:0]    |            |           |           |  |  |  |  |  |
|         |                  | _             |                 | _          | _                 |            | _          |           |           |  |  |  |  |  |
|         | ECNTBF16         |               |                 |            | ECNTBF            | 16[15:8]   |            |           |           |  |  |  |  |  |
|         |                  |               |                 |            | ECNTB             | F16[7:0]   |            |           |           |  |  |  |  |  |
|         | GRBF16           |               |                 |            | GRBF1             | 6[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |               | GRBF16[7:0]     |            |                   |            |            |           |           |  |  |  |  |  |
|         | ECNTCF16         |               | ECNTCF16[23:16] |            |                   |            |            |           |           |  |  |  |  |  |
|         |                  |               | ECNTCF16[15:8]  |            |                   |            |            |           |           |  |  |  |  |  |
|         |                  |               |                 |            | ECNTC             | F16[7:0]   |            |           |           |  |  |  |  |  |
|         |                  | _             |                 |            |                   |            |            |           | _         |  |  |  |  |  |
|         | GRAF16           | GRAF16[23:16] |                 |            |                   |            |            |           |           |  |  |  |  |  |
|         |                  |               | GRAF16[15:8]    |            |                   |            |            |           |           |  |  |  |  |  |
|         |                  |               |                 |            | GRAF1             | 16[7:0]    |            |           |           |  |  |  |  |  |
|         |                  | _             |                 | _          |                   |            | _          |           | _         |  |  |  |  |  |
|         | CDRF16           | CDRF16[23:16] |                 |            |                   |            |            |           |           |  |  |  |  |  |
|         |                  |               |                 |            | CDRF1             | 6[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |               | 1               | 1          | CDRF1             | 16[7:0]    | 1          | 1         | 1         |  |  |  |  |  |
|         |                  | _             | _               | _          |                   | _          | _          |           | _         |  |  |  |  |  |
|         | GRCF16           |               |                 |            | GRCF16            | 6[23:16]   |            |           |           |  |  |  |  |  |
|         |                  |               |                 |            | GRCF1             | 6[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |               |                 |            | GRCF <sup>-</sup> | 16[7:0]    |            |           |           |  |  |  |  |  |
|         |                  | _             | _               | _          | _                 | _          | _          | _         | _         |  |  |  |  |  |
|         | TCRF17           | С             | KSELF17[2:      | 0]         |                   | MDF17[2:0] | 1          | EGSELF    | =17[1:0]  |  |  |  |  |  |
|         | TIERF17          | _             | _               | _          | _                 | OVECF17    | OVEBF17    | OVEAF17   | ICEF17    |  |  |  |  |  |
|         | TSRF17           | <u></u>       |                 |            |                   | OVFCF17    | OVFBF17    | OVFAF17   | ICFF17    |  |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6     | 29/21/13/5 | 28/20/12/4        | 27/19/11/3   | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|---------------|----------------|------------|-------------------|--------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | ECNTAF17         |               | <u>I</u>       | <u>I</u>   | ECNTAF:           | 17[23:16]    |            |           | <u>,L</u> |  |  |  |  |
|         |                  |               |                |            | ECNTAF            | 17[15:8]     |            |           |           |  |  |  |  |
|         |                  |               |                |            | ECNTA             | F17[7:0]     |            |           |           |  |  |  |  |
|         |                  | _             | _              | _          | _                 | _            | _          | _         | _         |  |  |  |  |
|         | ECNTBF17         |               |                |            | ECNTBF            | 17[15:8]     |            |           |           |  |  |  |  |
|         |                  |               |                |            | ECNTB             | F17[7:0]     |            |           |           |  |  |  |  |
|         | GRBF17           |               |                |            | GRBF1             | 7[15:8]      |            |           |           |  |  |  |  |
|         |                  |               | GRBF17[7:0]    |            |                   |              |            |           |           |  |  |  |  |
|         | ECNTCF17         |               |                |            | ECNTCF            | 17[23:16]    |            |           |           |  |  |  |  |
|         |                  |               | ECNTCF17[15:8] |            |                   |              |            |           |           |  |  |  |  |
|         |                  |               |                |            | ECNTCI            | F17[7:0]     |            |           |           |  |  |  |  |
|         |                  | _             |                |            |                   |              |            |           | _         |  |  |  |  |
|         | GRAF17           |               |                |            |                   |              |            |           |           |  |  |  |  |
|         |                  |               | GRAF17[15:8]   |            |                   |              |            |           |           |  |  |  |  |
|         |                  |               | 1              | 1          | GRAF-             | 17[7:0]      | 1          | 1         |           |  |  |  |  |
|         |                  | _             |                |            |                   |              |            |           | _         |  |  |  |  |
|         | CDRF17           | CDRF17[23:16] |                |            |                   |              |            |           |           |  |  |  |  |
|         |                  |               | CDRF17[15:8]   |            |                   |              |            |           |           |  |  |  |  |
|         |                  |               | I              | I          | CDRF <sup>-</sup> | 17[7:0]<br>T | 1          | 1         | Т         |  |  |  |  |
|         |                  | _             |                |            |                   |              |            | _         |           |  |  |  |  |
|         | GRCF17           |               | GRCF17[23:16]  |            |                   |              |            |           |           |  |  |  |  |
|         |                  |               |                |            | GRCF1             |              |            |           |           |  |  |  |  |
|         |                  |               | T              | T          | GRCF <sup>-</sup> | 17[7:0]<br>T | 1          | 1         | Т         |  |  |  |  |
|         | T00540           | -             |                |            |                   |              |            |           |           |  |  |  |  |
|         | TCRF18           | C             | KSELF18[2:     | 0 <u>]</u> |                   | MDF18[2:0]   |            |           | F18[1:0]  |  |  |  |  |
|         | TIERF18          |               | _              |            | _                 | OVECF18      | -          | -         | ICEF18    |  |  |  |  |
|         | TSRF18           |               |                |            | FONTAS:           | OVFCF18      | OALRE 18   | OVFAF18   | ICFF18    |  |  |  |  |
|         | ECNTAF18         |               |                |            | ECNTAF            |              |            |           |           |  |  |  |  |
|         |                  |               |                |            | ECNTAF            |              |            |           |           |  |  |  |  |
|         |                  |               | ECNTAF18[7:0]  |            |                   |              |            |           |           |  |  |  |  |
|         |                  |               |                |            |                   | _            |            |           |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7  | 30/22/14/6    | 29/21/13/5 | 28/20/12/4          | 27/19/11/3    | 26/18/10/2 | 25/17/9/1 | 24/16/8/0     |  |  |  |  |  |
|---------|------------------|-------------|---------------|------------|---------------------|---------------|------------|-----------|---------------|--|--|--|--|--|
| ATU-III | ECNTBF18         |             |               |            | ECNTBF              | 18[15:8]      |            |           |               |  |  |  |  |  |
|         |                  |             |               |            | ECNTB               | -18[7:0]      |            |           |               |  |  |  |  |  |
|         | GRBF18           |             |               |            | GRBF1               | 8[15:8]       |            |           |               |  |  |  |  |  |
|         |                  | GRBF18[7:0] |               |            |                     |               |            |           |               |  |  |  |  |  |
|         | ECNTCF18         |             |               |            | ECNTCF              | 18[23:16]     |            |           |               |  |  |  |  |  |
|         |                  |             |               |            | ECNTCF              | 18[15:8]      |            |           |               |  |  |  |  |  |
|         |                  |             |               |            | ECNTC               | =18[7:0]      |            |           |               |  |  |  |  |  |
|         |                  | _           |               |            |                     |               |            |           |               |  |  |  |  |  |
|         | GRAF18           |             |               |            | GRAF18              | 3[23:16]      |            |           |               |  |  |  |  |  |
|         |                  |             |               |            | GRAF1               | 8[15:8]       |            |           |               |  |  |  |  |  |
|         |                  |             |               |            | GRAF*               | 18[7:0]       |            |           |               |  |  |  |  |  |
|         |                  | _           |               | _          | _                   |               | _          | _         |               |  |  |  |  |  |
|         | CDRF18           |             | CDRF18[23:16] |            |                     |               |            |           |               |  |  |  |  |  |
|         |                  |             | CDRF18[15:8]  |            |                     |               |            |           |               |  |  |  |  |  |
|         |                  |             | CDRF18[7:0]   |            |                     |               |            |           |               |  |  |  |  |  |
|         |                  | _           | _             | _          | _                   | _             | _          | _         | _             |  |  |  |  |  |
|         | GRCF18           |             |               |            | GRCF18              | 3[23:16]      |            |           |               |  |  |  |  |  |
|         |                  |             |               |            | GRCF1               | 8[15:8]       |            |           |               |  |  |  |  |  |
|         |                  |             | T             | T          | GRCF <sup>-</sup>   | 18[7:0]       | Т          | T         | T             |  |  |  |  |  |
|         |                  | _           |               | _          | _                   |               |            | _         | _             |  |  |  |  |  |
|         | TCRF19           | С           | KSELF19[2:    | 0]         |                     | MDF19[2:0]    | Т          | EGSELI    | =19[1:0]<br>- |  |  |  |  |  |
|         | TIERF19          |             | _             | _          | _                   | OVECF19       | OVEBF19    | OVEAF19   | ICEF19        |  |  |  |  |  |
|         | TSRF19           | _           |               |            |                     | OVFCF9        | OVFBF9     | OVFAF9    | ICFF9         |  |  |  |  |  |
|         | ECNTAF19         |             |               |            | ECNTAF <sup>-</sup> | 19[23:16]     |            |           |               |  |  |  |  |  |
|         |                  |             |               |            | ECNTAF              |               |            |           |               |  |  |  |  |  |
|         |                  |             | ı             | ı          | ECNTA               | =19[7:0]<br>T | T          | ı         | 1             |  |  |  |  |  |
|         |                  |             |               |            |                     |               |            |           |               |  |  |  |  |  |
|         | ECNTBF19         |             |               |            | ECNTBF              |               |            |           |               |  |  |  |  |  |
|         |                  |             |               |            | ECNTBI              | -19[7:0]      |            |           |               |  |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7    | 30/22/14/6    | 29/21/13/5 | 28/20/12/4        | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|---------------|---------------|------------|-------------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | GRBF19           |               |               | •          | GRBF1             | 9[15:8]    | •          | •         | II.       |  |  |  |  |
|         |                  |               |               |            | GRBF <sup>-</sup> | 19[7:0]    |            |           |           |  |  |  |  |
|         | ECNTCF19         |               |               |            | ECNTCF            | 19[23:16]  |            |           |           |  |  |  |  |
|         |                  |               |               |            | ECNTCF            | 19[15:8]   |            |           |           |  |  |  |  |
|         |                  |               |               |            | ECNTC             | F19[7:0]   |            |           |           |  |  |  |  |
|         |                  | _             | _             | _          | _                 | _          | _          |           |           |  |  |  |  |
|         | GRAF19           |               |               |            | GRAF19            | 9[23:16]   |            |           |           |  |  |  |  |
|         |                  |               |               |            | GRAF1             | 9[15:8]    |            |           |           |  |  |  |  |
|         |                  |               |               |            | GRAF <sup>-</sup> | 19[7:0]    |            |           |           |  |  |  |  |
|         |                  | —             | _             | _          | _                 |            | _          |           |           |  |  |  |  |
|         | CDRF19           |               | CDRF19[23:16] |            |                   |            |            |           |           |  |  |  |  |
|         |                  |               | CDRF19[15:8]  |            |                   |            |            |           |           |  |  |  |  |
|         |                  |               |               |            | CDRF <sup>-</sup> | 19[7:0]    |            |           |           |  |  |  |  |
|         |                  | _             | _             | _          | _                 | _          | _          | _         | _         |  |  |  |  |
|         | GRCF19           | GRCF19[23:16] |               |            |                   |            |            |           |           |  |  |  |  |
|         |                  | GRCF19[15:8]  |               |            |                   |            |            |           |           |  |  |  |  |
|         |                  |               |               |            | GRCF              | 19[7:0]    |            |           |           |  |  |  |  |
|         |                  | _             | _             | _          | _                 |            | _          |           |           |  |  |  |  |
|         | TSTRG            | _             | _             | STRG5      | STRG              | STRG3      | STRG2      | STRG1     | STRG0     |  |  |  |  |
|         | TCRG0            | _             | C             | KSELG0[2:  | 0]                |            |            | CMPOEG0   | CMEG0     |  |  |  |  |
|         | TSRG0            | _             | _             |            | _                 |            |            | OVFG0     | CMFG0     |  |  |  |  |
|         | TCNTG0           |               |               |            | TCNTG             | i0[15:8]   |            |           |           |  |  |  |  |
|         |                  |               |               |            | TCNT              | G0[7:0]    |            |           |           |  |  |  |  |
|         | OCRG0            |               |               |            | OCRG              | 0[15:8]    |            |           |           |  |  |  |  |
|         |                  |               | OCRG0[7:0]    |            |                   |            |            |           |           |  |  |  |  |
|         | TCRG1            | _             | C             | KSELG1[2:  | 0]                | _          | _          | CMPOEG1   | CMEG1     |  |  |  |  |
|         | TSRG1            | _             |               |            | _                 |            | _          | OVFG1     | CMFG1     |  |  |  |  |
|         | TCNTG1           |               |               |            | TCNTG             | 1[15:8]    |            |           |           |  |  |  |  |
|         |                  |               |               |            | TCNT              | G1[7:0]    |            |           |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7   | 30/22/14/6                               | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|--------------|--|------------|------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | OCRG1            |              | I .                                      | I .        | OCRG       | 1[15:8]    |            | l         | _!        |  |  |  |  |
|         |                  |              |  |            | OCRG       | 1[7:0]     |            |           |           |  |  |  |  |
|         | TCRG2            | _            | C  | CKSELG2[2: | 0]         | _          | _          | CMPOEG2   | CMEG2     |  |  |  |  |
|         | TSRG2            | _            | _  |            |            | _          | _          | OVFG2     | CMFG2     |  |  |  |  |
|         | TCNTG2           |              | TCNTG2[15:8]                             |            |            |            |            |           |           |  |  |  |  |
|         |                  |              | TCNTG2[7:0]                              |            |            |            |            |           |           |  |  |  |  |
|         | OCRG2            |              |  |            | OCRG       | 2[15:8]    |            |           |           |  |  |  |  |
|         |                  |              | OCRG2[7:0]  CKSELG3[2:0] — — CMPOEG3 CME |            |            |            |            |           |           |  |  |  |  |
|         | TCRG3            | _            | — CKSELG3[2:0] — — СмРОЕG3               |            |            |            |            |           |           |  |  |  |  |
|         | TSRG3            | _            | _  | _          | _          | _          |            | OVFG3     | CMFG3     |  |  |  |  |
|         | TCNTG3           |              | TCNTG3[15:8]                             |            |            |            |            |           |           |  |  |  |  |
|         |                  |              |  |            | TCNT       | G3[7:0]    |            |           |           |  |  |  |  |
|         | OCRG3            |              | OCRG3[15:8]                              |            |            |            |            |           |           |  |  |  |  |
|         |                  |              | OCRG3[7:0]                               |            |            |            |            |           |           |  |  |  |  |
|         | TCRG4            | _            | C  | CKSELG4[2: | 0]         | _          |            | CMPOEG4   | CMEG4     |  |  |  |  |
|         | TSRG4            | _            | _  | _          | _          | _          | _          | OVFG4     | CMFG4     |  |  |  |  |
|         | TCNTG4           |              | TCNTG4[15:8]                             |            |            |            |            |           |           |  |  |  |  |
|         |                  |              |  |            | TCNT       | G4[7:0]    |            |           |           |  |  |  |  |
|         | OCRG4            |              |  |            | OCRG-      | 4[15:8]    |            |           |           |  |  |  |  |
|         |                  |              |  |            | OCRG       | 4[7:0]     |            |           |           |  |  |  |  |
|         | TCRG5            | _            | CMPOEG5                                  | CMEG5      |            |            |            |           |           |  |  |  |  |
|         | TSRG5            | _            |  | _          | OVFG5      | CMFG5      |            |           |           |  |  |  |  |
|         | TCNTG5           |              | TCNTG5[15:8]                             |            |            |            |            |           |           |  |  |  |  |
|         |                  |              |  |            | TCNT       | G5[7:0]    |            |           |           |  |  |  |  |
|         | OCRG5            |              |  |            | OCRG:      | 5[15:8]    |            |           |           |  |  |  |  |
|         |                  | OCRG5[7:0]   |  |            |            |            |            |           |           |  |  |  |  |
|         | TCRH             | CKSELH0[2:0] |  |            |            |            |            |           |           |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7 | 30/22/14/6    | 29/21/13/5  | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |  |
|---------|------------------|------------|---------------|-------------|------------|------------|------------|-----------|-----------|--|--|--|--|--|
| ATU-III | TSRH             | _          | _             | _           | _          | _          | OVF2H      | OVF1H     | CMFH      |  |  |  |  |  |
|         | TCNT1H           |            |               |             | TCNT1      | H[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |            |               |             | TCNT1      | H[7:0]     |            |           |           |  |  |  |  |  |
|         | OCR1H            |            |               |             | OCR1F      | H[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |            |               |             | OCR1       | H[7:0]     |            |           |           |  |  |  |  |  |
|         | TCNT2H           |            |               |             | TCNT2F     | 1[31:24]   |            |           |           |  |  |  |  |  |
|         |                  |            | TCNT2H[23:16] |             |            |            |            |           |           |  |  |  |  |  |
|         |                  |            | TCNT2H[15:8]  |             |            |            |            |           |           |  |  |  |  |  |
|         |                  |            |               |             | TCNT2      | 2H[7:0]    |            |           |           |  |  |  |  |  |
|         | TSTRJ            | _          | _             | _           | _          | _          | _          | STRJ1     | STRJ0     |  |  |  |  |  |
|         | TCRJ0            | _          | (             | CKSELJ0[2:0 | )]         | _          | NCEJ0      | _         | _         |  |  |  |  |  |
|         | FCRJ0            | FIFOENJ0   | _             | FVCRENJ0    | FRSTJ0     | _          | _          | FDFTRO    | 3J0[1:0]  |  |  |  |  |  |
|         | TSRJ0            | _          | _             | _           | FVLDFJ0    | CMFJ0      | OVFJ0      | FDOVFJ0   | FDFFJ0    |  |  |  |  |  |
|         | TIERJ0           | _          | _             | _           | _          | _          | OVEJ0      | FDOVEJ0   | FDFEJ0    |  |  |  |  |  |
|         | FDNRJ0           | _          | _             | _           | _          |            | FDNJ0[3:0] |           |           |  |  |  |  |  |
|         | NCNTJ0           |            | NCCNTJ0[7:0]  |             |            |            |            |           |           |  |  |  |  |  |
|         | NCRJ0            |            | NCTJ0[7:0]    |             |            |            |            |           |           |  |  |  |  |  |
|         | TCNTJ0           |            |               |             | TCNTJ      | 0[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |            |               |             | TCNT       | J0[7:0]    |            |           |           |  |  |  |  |  |
|         | OCRJ0            |            |               |             | OCRJ       | )[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |            |               |             | OCRJ       | 0[7:0]     |            |           |           |  |  |  |  |  |
|         | FIFOJ0           |            |               |             | FIFODJ     | 0[15:8]    |            |           |           |  |  |  |  |  |
|         |                  |            |               |             | FIFOD      | J0[7:0]    |            |           |           |  |  |  |  |  |
|         | TCRJ1            |            | (             | CKSELJ1[2:0 | )]         |            | NCEJ1      | IOJ1      | [1:0]     |  |  |  |  |  |
|         | FCRJ1            | FIFOENJ1   | _             | FVCRENJ1    | FRSTJ1     | _          | _          | FDFTRO    | 3J1[1:0]  |  |  |  |  |  |
|         | TSRJ1            |            |               |             | FVLDFJ1    | CMFJ1      | OVFJ1      | FDOVFJ1   | FDFFJ1    |  |  |  |  |  |
|         | TIERJ1           |            |               |             |            |            | OVEJ1      | FDOVEJ1   | FDFEJ1    |  |  |  |  |  |
|         | FDNRJ1           | _          |               |             |            |            | FDNJ       | 1[3:0]    |           |  |  |  |  |  |

| Module  | Register<br>Name | 31/23/15/7  | 30/22/14/6    | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------|------------------|-------------|---------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|
| ATU-III | NCNTJ1           |             |               |            | NCCNT      |            |            |           |           |  |  |  |  |
|         | NCRJ1            |             |               |            | NCTJ       |            |            |           |           |  |  |  |  |
|         | TCNTJ1           |             |               |            | TCNTJ      | 1[15:8]    |            |           |           |  |  |  |  |
|         |                  |             |               |            | TCNTJ      | 11[7:0]    |            |           |           |  |  |  |  |
|         | OCRJ1            |             | OCRJ1[15:8]   |            |            |            |            |           |           |  |  |  |  |
|         |                  |             | OCRJ1[7:0]    |            |            |            |            |           |           |  |  |  |  |
|         | FIFOJ1           |             | FIFODJ1[15:8] |            |            |            |            |           |           |  |  |  |  |
|         |                  |             | FIFODJ1[7:0]  |            |            |            |            |           |           |  |  |  |  |
| WDT     | WTCR             |             | TCRKEY[7:0]   |            |            |            |            |           |           |  |  |  |  |
| l       |                  | _           | WT/ĪT         | TME        | _          |            |            | CKS[2:0]  |           |  |  |  |  |
|         | WTCNT            |             | •             | •          | TCNTK      | EY[7:0]    | •          |           |           |  |  |  |  |
|         |                  |             |               |            | TCNT       | [7:0]      |            |           |           |  |  |  |  |
|         | WTSR             |             |               |            | TSRKE      | Y[7:0]     |            |           |           |  |  |  |  |
|         | WPCP             | WOVF        | _             | _          | _          | IOVF       | _          | _         | _         |  |  |  |  |
|         | WRCR             | RCRKEY[7:0] |               |            |            |            |            |           |           |  |  |  |  |
|         | Whoh             | RSTE        | _             | _          | _          | _          | _          | _         | _         |  |  |  |  |
| CMT     | CMSTR            | _           | _             | _          | _          | _          | _          | _         | _         |  |  |  |  |
|         |                  | _           | _             | _          | _          | _          | _          | STR1      | STR0      |  |  |  |  |
|         | CMCR_0           | _           | CMIE          | _          | _          | _          | _          | CKS       | [1:0]     |  |  |  |  |
|         | CMSR_0           | _           | _             | _          | _          |            | _          | _         | CMF       |  |  |  |  |
|         | CMCNT_0          |             |               |            |            |            |            |           |           |  |  |  |  |
|         |                  |             |               |            |            |            |            |           |           |  |  |  |  |
|         | CMCOR_0          |             |               |            |            |            |            |           |           |  |  |  |  |
|         |                  |             |               |            |            |            |            |           |           |  |  |  |  |
|         | CMCR_1           | _           | CMIE          | _          |            | _          | _          | CKS       | [1:0]     |  |  |  |  |
|         | CMSR_1           |             | _             | _          |            |            | _          | _         | CMF       |  |  |  |  |
|         | CMCNT_1          |             |               |            |            |            |            |           |           |  |  |  |  |
|         |                  |             |               |            |            |            |            |           |           |  |  |  |  |
| İ       | CMCOR_1          |             |               |            |            |            |            |           |           |  |  |  |  |
|         |                  |             |               |            |            |            |            |           |           |  |  |  |  |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| SCI    | SCSMR1A          | C/Ā        | CHR        | PE         | O/Ē        | STOP       | _          | CKS       | 5[1:0]    |
|        | SCBRR1A          |            |            |            |            |            |            |           |           |
|        | SCSCR1A          | TIE        | RIE        | TE         | RE         | _          | TEIE       | CKE1      | _         |
|        | SCTDR1A          |            |            |            |            |            |            |           |           |
|        | SCSSR1A          | TDRE       | RDRF       | ORER       | FER        | PER        | TEND       | _         |           |
|        | SCRDR1A          |            |            |            |            |            |            |           |           |
|        | SCSMR1B          | C/Ā        | CHR        | PE         | O/E        | STOP       | _          | CKS       | 5[1:0]    |
|        | SCBRR1B          |            |            |            |            |            |            |           |           |
|        | SCSCR1B          | TIE        | RIE        | TE         | RE         | _          | TEIE       | CKE1      |           |
|        | SCTDR1B          |            |            |            |            |            |            |           |           |
|        | SCSSR1B          | TDRE       | RDRF       | ORER       | FER        | PER        | TEND       | _         | _         |
|        | SCRDR1B          |            |            |            |            |            |            |           |           |
|        | SCSMR1C          | C/Ā        | CHR        | PE         | O/E        | STOP       | _          | CKS       | 5[1:0]    |
|        | SCBRR1C          |            |            |            |            |            |            |           |           |
|        | SCSCR1C          | TIE        | RIE        | TE         | RE         | _          | TEIE       | CKE1      | _         |
|        | SCTDR1C          |            |            |            |            |            |            |           |           |
|        | SCSSR1C          | TDRE       | RDRF       | ORER       | FER        | PER        | TEND       | _         |           |
|        | SCRDR1C          |            |            |            |            |            |            |           |           |
|        | SCSMR1D          | C/Ā        | CHR        | PE         | O/E        | STOP       | _          | CKS       | 5[1:0]    |
|        | SCBRR1D          |            |            |            |            |            |            |           |           |
|        | SCSCR1D          | TIE        | RIE        | TE         | RE         | _          | TEIE       | CKE1      | _         |
|        | SCTDR1D          |            |            |            |            |            |            |           |           |
|        | SCSSR1D          | TDRE       | RDRF       | ORER       | FER        | PER        | TEND       | _         |           |
|        | SCRDR1D          |            |            |            |            |            |            |           |           |
|        | SCSMR1E          | C/Ā        | CHR        | PE         | O/E        | STOP       | _          | CKS       | 5[1:0]    |
|        | SCBRR1E          |            |            |            |            |            |            |           |           |
|        | SCSCR1E          | TIE        | RIE        | TE         | RE         | _          | TEIE       | CKE1      | _         |
|        | SCTDR1E          |            |            |            |            |            |            |           |           |
|        | SCSSR1E          | TDRE       | RDRF       | ORER       | FER        | PER        | TEND       | _         |           |
|        | SCRDR1E          | 1          |            |            |            |            |            |           |           |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3        | 26/18/10/2              | 25/17/9/1  | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|-------------------|-------------------------|------------|-----------|
| RSPI   | SPCRA            | SPRIE      | SPE        | SPTIE      | SPEIE      | MSTR              | MODFEN                  | _          | _         |
|        | SSLPA            | SSL7P      | SSL6P      | SSL5P      | SSL4P      | SSL3P             | SSL2P                   | SSL1P      | SSL0P     |
|        | SPPCRA           | _          | _          | MOIFE      | MOIFV      | _                 | SPOM                    | _          | SPLP      |
|        | SPSRA            | SPRF       | _          | SPTEF      | _          | _                 | MODF                    | _          | OVRF      |
|        | SPDRA            | SPD15      | SPD14      | SPD13      | SPD12      | SPD11             | SPD10                   | SPD9       | SPD8      |
|        |                  | SPD7       | SPD6       | SPD5       | SPD4       | SPD3              | SPD2                    | SPD1       | SPD0      |
|        | SPSCRA           | _          | _          | _          | _          | _                 |                         | SPSLN[2:0] |           |
|        | SPSSRA           | _          |            | SPECM[2:0  | ]          | _                 |                         | SPCP[2:0]  |           |
|        | SPBRA            | SPR7       | SPR6       | SPR5       | SPR4       | SPR3              | SPR2                    | SPR1       | SPR0      |
|        | SPCKDA           | _          | _          | _          | _          | _                 |                         | SCKDL[2:0] |           |
|        | SSLNDA           | _          | _          | _          | _          | _                 |                         | SLNDL[2:0] |           |
|        | SPNDA            | _          | _          | _          | _          | _                 |                         | SPNDL[2:0] |           |
|        | SPCMDA0          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |                   | SPB[3:0]<br>V[1:0] CPOL | [3:0]      |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD               |                         | CPOL       | СРНА      |
|        | SPCMDA1          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |                   | SPB[3:0]                |            |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRDV[1:0] CPOL CF |                         |            | СРНА      |
|        | SPCMDA2          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       | SPB[3:0]          |                         |            |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRDV[1:0] CPOL C  |                         |            | СРНА      |
|        | SPCMDA3          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |                   | SPB                     | [3:0]      |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD               | V[1:0]                  | CPOL       | СРНА      |
|        | SPCMDA4          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |                   | SPB                     | [3:0]      |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD               | V[1:0]                  | CPOL       | СРНА      |
|        | SPCMDA5          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |                   | SPB                     | [3:0]      |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD               | V[1:0]                  | CPOL       | СРНА      |
|        | SPCMDA6          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |                   | SPB                     | [3:0]      |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD               | V[1:0]                  | CPOL       | СРНА      |
|        | SPCMDA7          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |                   | SPB                     | [3:0]      |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD               | OV[1:0] CPOL CPH        |            | СРНА      |
|        | SPCRB            | SPRIE      | SPE        | SPTIE      | SPEIE      | MSTR              | MODFEN                  | _          |           |
|        | SSLPB            | SSL7P      | SSL6P      | SSL5P      | SSL4P      | SSL3P             | SSL2P                   | SSL1P      | SSL0P     |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2              | 25/17/9/1  | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|-------------------------|------------|-----------|
| RSPI   | SPPCRB           |            |            | MOIFE      | MOIFV      | _          | SPOM                    |            | SPLP      |
|        | SPSRB            | SPRF       |            | SPTEF      | _          | _          | MODF                    | _          | OVRF      |
|        | SPDRB            | SPD15      | SPD14      | SPD13      | SPD12      | SPD11      | SPD10                   | SPD9       | SPD8      |
|        |                  | SPD7       | SPD6       | SPD5       | SPD4       | SPD3       | SPD2                    | SPD1       | SPD0      |
|        | SPSCRB           | _          | _          | _          | _          | _          |                         | SPSLN[2:0] |           |
|        | SPSSRB           | _          |            | SPECM[2:0  | ]          | _          |                         | SPCP[2:0]  |           |
|        | SPBRB            | SPR7       | SPR6       | SPR5       | SPR4       | SPR3       | SPR2                    | SPR1       | SPR0      |
|        | SPCKDB           | _          | _          | _          | _          | _          |                         | SCKDL[2:0] |           |
|        | SSLNDB           | _          | _          | _          | _          | _          |                         | SLNDL[2:0] |           |
|        | SPNDB            | _          | _          | _          | _          | _          | SPNDL[2:0]              |            |           |
|        | SPCMDB0          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB[3:0]<br>V[1:0] CPOL |            |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        |                         | CPOL       | СРНА      |
|        | SPCMDB1          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       | SPB[3:0]   |                         |            | •         |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]                  | CPOL       | СРНА      |
|        | SPCMDB2          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB                     | [3:0]      | •         |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | BRDV[1:0]               |            | СРНА      |
|        | SPCMDB3          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       | SPB[3:0]   |                         |            |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]                  | CPOL       | СРНА      |
|        | SPCMDB4          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB                     | [3:0]      |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]                  | CPOL       | СРНА      |
|        | SPCMDB5          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB                     | [3:0]      |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]                  | CPOL       | СРНА      |
|        | SPCMDB6          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB                     | [3:0]      |           |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD'       | V[1:0]                  | CPOL       | СРНА      |
|        | SPCMDB7          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB                     | [3:0]      | •         |
|        |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]                  | CPOL       | СРНА      |
|        | SPCRC            | SPRIE      | SPE        | SPTIE      | SPEIE      | MSTR       | MODFEN                  | _          | _         |
|        | SSLPC            | SSL7P      | SSL6P      | SSL5P      | SSL4P      | SSL3P      | SSL2P                   | SSL1P      | SSL0P     |
|        | SPPCRC           | _          | _          | MOIFE      | MOIFV      | _          | SPOM                    | _          | SPLP      |
|        | SPSRC            | SPRF       | _          | SPTEF      | _          | _          | MODF                    | _          | OVRF      |

| Module   | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2   | 25/17/9/1   | 24/16/8/0 |  |
|----------|------------------|------------|------------|------------|------------|------------|--------------|---|-----------|--|
| RSPI     | SPDRC            | SPD15      | SPD14      | SPD13      | SPD12      | SPD11      | SPD10        | SPD9  | SPD8      |  |
|          |                  | SPD7       | SPD6       | SPD5       | SPD4       | SPD3       | SPD2         | SPD1  | SPD0      |  |
|          | SPSCRC           |            |            |            |            |            |              | SPSLN[2:0]  | ı         |  |
|          | SPSSRC           | _          |            | SPECM[2:0] | ]          |            |              | SPCP[2:0]   |           |  |
|          | SPBRC            | SPR7       | SPR6       | SPR5       | SPR4       | SPR3       | SPR2         | SPR1  | SPR0      |  |
|          | SPCKDC           | _          | _          | _          | _          | _          |              | SCKDL[2:0]  SLNDL[2:0]  SPNDL[2:0]  SPB[3:0]  [1:0]  CPOL |           |  |
|          | SSLNDC           | _          | _          | _          |            |            |              |   |           |  |
|          | SPNDC            | _          | _          | _          |            |            |              |   |           |  |
|          | SPCMDC0          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB          |   |           |  |
|          |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]       |   |           |  |
|          | SPCMDC1          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB          | [3:0]   |           |  |
|          |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]       | СРНА  |           |  |
|          | SPCMDC2          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB          | SPB[3:0]  |           |  |
|          |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]       | CPOL  | СРНА      |  |
|          | SPCMDC3          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB          | [3:0]   |           |  |
|          |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]       | CPOL  | СРНА      |  |
|          | SPCMDC4          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            |              | [3:0]   | ı         |  |
|          |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]       | CPOL  | СРНА      |  |
|          | SPCMDC5          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB          | [3:0]   |           |  |
|          |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]       | CPOL  | СРНА      |  |
|          | SPCMDC6          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB          | [3:0]   |           |  |
|          |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]       | CPOL  | СРНА      |  |
|          | SPCMDC7          | SCKDEN     | SLNDEN     | SPNDEN     | LSBF       |            | SPB          | [3:0]   |           |  |
|          |                  | SSLKP      |            | SSLA[2:0]  |            | BRD        | V[1:0]       | CPOL  | СРНА      |  |
| RCAN-TL1 | MCR              | MCR15      | MCR14      | _          | _          |            |              | TST[2:0]  |           |  |
| (RCAN_A) |                  | MCR7       | MCR6       | MCR5       | _          | _          | MCR2 MCR1    |   | MCR0      |  |
|          | GSR              | _          | _          | _          | _          | _          |              |   | _         |  |
|          |                  | _          | _          | GSR5       | GSR4       | GSR3       | GSR2 GSR1 GS |   | GSR0      |  |
|          | BCR1             |            | TSG        | 1[3:0]     | •          | _          |              | TSG2[2:0]   |           |  |
|          |                  | _          |            | SJW        | /[1:0]     | _          | _            | _   | BSP       |  |

| Module   | Register<br>Name | 31/23/15/7  | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |  |
|----------|------------------|-------------|--------------|------------|------------|------------|------------|-----------|-----------|--|--|--|--|--|
| RCAN-TL1 | BCR0             | _           | _            | _          |            | _          | _          | _         | _         |  |  |  |  |  |
| (RCAN_A) |                  |             | I.           | I.         | BRP        | [7:0]      | I.         | I.        | 1         |  |  |  |  |  |
|          | IRR              | IRR15       | IRR14        | IRR13      | IRR12      | IRR11      | IRR10      | 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            |             |              |            | TXPR1      | I[15:8]    |            |           | •         |  |  |  |  |  |
|          |                  |             | TXPR1[7:0]   |            |            |            |            |           |           |  |  |  |  |  |
|          | TXPR0            |             | TXPR0[15:8]  |            |            |            |            |           |           |  |  |  |  |  |
|          |                  |             | TXPR0[7:1] — |            |            |            |            |           |           |  |  |  |  |  |
|          | TXCR1            | TXCR1[15:8] |              |            |            |            |            |           |           |  |  |  |  |  |
|          |                  |             | TXCR1[7:0]   |            |            |            |            |           |           |  |  |  |  |  |
|          | TXCR0            |             | TXCR0[15:8]  |            |            |            |            |           |           |  |  |  |  |  |
|          |                  |             |              |            | TXCR0[7:1] |            |            |           | _         |  |  |  |  |  |
|          | TXACK1           |             | TXACK1[15:8] |            |            |            |            |           |           |  |  |  |  |  |
|          |                  |             |              |            | TXAC       | (1[7:0]    |            |           |           |  |  |  |  |  |
|          | TXACK0           |             |              |            | TXACK      | 0[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |             |              |            | TXACK0[7:1 | ]          |            |           | _         |  |  |  |  |  |
|          | ABACK1           |             |              |            | ABACK      | 1[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |             |              |            | ABAC       | K1[7:0]    |            |           |           |  |  |  |  |  |
|          | ABACK0           |             |              |            | ABACK      | 0[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |             |              |            | ABACK0[7:1 | ]          |            |           | _         |  |  |  |  |  |
|          | RXPR1            |             |              |            | RXPR1      | 1[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |             |              |            | RXPR       | 1[7:0]     |            |           |           |  |  |  |  |  |
|          | RXPR0            |             |              |            | RXPR       | )[15:8]    |            |           |           |  |  |  |  |  |
|          |                  | RXPR0[7:0]  |              |            |            |            |            |           |           |  |  |  |  |  |
|          | RFPR1            |             |              |            | RFPR1      | 1[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |             |              |            | RFPR       | 1[7:0]     |            |           |           |  |  |  |  |  |

| Module   | Register<br>Name | 31/23/15/7 | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |
|----------|------------------|------------|--------------|------------|------------|------------|------------|-----------|-----------|--|--|--|
| RCAN-TL1 | RFPR0            |            |              |            | RFPR(      |            |            |           | ]         |  |  |  |
| (RCAN_A) |                  |            |              |            | RFPR       | 0[7:0]     |            |           |           |  |  |  |
|          | MBIMR1           |            |              |            | MBIMR      | 1[15:8]    |            |           |           |  |  |  |
|          |                  |            |              |            | MBIMF      | R1[7:0]    |            |           |           |  |  |  |
|          | MBIMR0           |            |              |            | MBIMR      | 10[15:8]   |            |           |           |  |  |  |
|          |                  |            |              |            | MBIMF      | R0[7:0]    |            |           |           |  |  |  |
|          | UMSR1            |            |              |            | UMSR       | 1[15:8]    |            |           |           |  |  |  |
|          |                  |            |              |            | UMSR       | R1[7:0]    |            |           |           |  |  |  |
|          | UMSR0            |            | UMSR0[15:8]  |            |            |            |            |           |           |  |  |  |
|          |                  |            |              | T          | UMSR       | R0[7:0]    |            |           | T         |  |  |  |
|          | TTCR0            | TCR15      | TCR14        | TCR13      | TCR12      | TCR11      | TCR10      | _         | _         |  |  |  |
|          |                  | _          | TCR6         | TPSC5      | TPSC4      | TPSC3      | TPSC2      | TPSC1     | TPSC0     |  |  |  |
|          | CMAX_TEW         | _          | _            |            | _          | _          |            | CMAX[2:0] |           |  |  |  |
|          |                  | _          |              |            |            |            | TEW        | [3:0]     |           |  |  |  |
|          | RFTROFF          |            | RFTROFF[7:0] |            |            |            |            |           |           |  |  |  |
|          |                  | _          | _            |            | _          | _          | _          | _         | _         |  |  |  |
|          | TSR              | _          |              | _          | _          | _          | _          |           | _         |  |  |  |
|          |                  | _          | _            | _          | TSR4       | TSR3       | TSR2       | TSR1      | TSR0      |  |  |  |
|          | CCR              | _          | _            |            |            | _          | _          | _         | _         |  |  |  |
|          |                  | _          |              |            |            | CCR        | [5:0]      |           |           |  |  |  |
|          | TCNTR            |            |              |            | TCNTF      | R[15:8]    |            |           |           |  |  |  |
|          |                  |            |              |            | TCNT       | R[7:0]     |            |           |           |  |  |  |
|          | CYCTR            |            |              |            | CYCTF      | R[15:8]    |            |           |           |  |  |  |
|          |                  |            |              |            | CYCT       | R[7:0]     |            |           |           |  |  |  |
|          | RFMK             |            |              |            | RFMK       | [15:8]     |            |           |           |  |  |  |
|          |                  |            |              |            | RFM        | ζ[7:0]     |            |           |           |  |  |  |
|          | TCMR0            |            |              |            | TCMR       | 0[15:8]    |            |           |           |  |  |  |
|          |                  |            | TCMR0[7:0]   |            |            |            |            |           |           |  |  |  |
|          | TCMR1            |            |              |            | TCMR       | 1[15:8]    |            |           |           |  |  |  |
|          |                  |            |              |            | TCMR       | R1[7:0]    |            |           |           |  |  |  |

| Register<br>Name    | 31/23/15/7  | 30/22/14/6                        | 29/21/13/5                     | 28/20/12/4        | 27/19/11/3 | 26/18/10/2  | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|---------------------|---|-----------------------------------|--------------------------------|-------------------|------------|-------------|-----------|-----------|--|--|--|--|
| TCMR2               |   | I                                 |                                | TCMR2             | 2[15:8]    |             |           | ı         |  |  |  |  |
|                     |   |                                   |                                | TCMR              | 2[7:0]     |             |           |           |  |  |  |  |
| TTTSEL              | _   |                                   |                                | Т                 | TTSEL[14:8 | ·]          |           |           |  |  |  |  |
|                     | _   | _                                 | _                              | _                 | _          | _           |           | _         |  |  |  |  |
| MB[0].              | IDE   | RTR                               | _                              |                   |            | STDID[10:6] |           |           |  |  |  |  |
| CONTROL0_H          |   |                                   | STDII                          | D[5:0]            |            |             | EXTID     | [17:16]   |  |  |  |  |
| MB[0].              |   |                                   |                                | EXTID             | [15:8]     |             |           |           |  |  |  |  |
| CONTROL0_L          |   |                                   |                                | EXTID             | D[7:0]     |             |           |           |  |  |  |  |
| MB[0].              | IDE_LAFM  | _                                 | _                              |                   | STE        | DID_LAFM[1  | 0:6]      |           |  |  |  |  |
| LAFM0               |   | STDID_LAFM[5:0] EXTID_LAFM[17:16] |                                |                   |            |             |           |           |  |  |  |  |
| MB[0].              |   | EXTID_LAFM[15:8]                  |                                |                   |            |             |           |           |  |  |  |  |
| LAFM1               |   | EXTID_LAFM[7:0]                   |                                |                   |            |             |           |           |  |  |  |  |
| MB[0].              |   | MSG_DATA_0                        |                                |                   |            |             |           |           |  |  |  |  |
| DATA_01             | MSG_DATA_1  |                                   |                                |                   |            |             |           |           |  |  |  |  |
| MB[0].              |   | MSG_DATA_2                        |                                |                   |            |             |           |           |  |  |  |  |
|                     |   |                                   |                                |                   |            |             |           |           |  |  |  |  |
|                     |   |                                   |                                |                   |            |             |           |           |  |  |  |  |
|                     |   |                                   |                                |                   |            |             |           |           |  |  |  |  |
|                     |   |                                   |                                |                   |            |             |           |           |  |  |  |  |
|                     |   | T                                 | I                              | MSG_D             | ATA_7      | I           |           |           |  |  |  |  |
|                     |   |                                   | NMC                            |                   |            |             |           |           |  |  |  |  |
| MDIO                | _   | _                                 | _                              |                   |            | DLC         | [3:0]     |           |  |  |  |  |
| мв[о].<br>TIMESTAMP |   |                                   |                                |                   |            |             |           |           |  |  |  |  |
| MD[4]               | IDE   | DTD                               |                                | Timesta           |            | CTDID(10.61 |           |           |  |  |  |  |
| CONTROLO_H          | IDE   | NIN                               | 21010[10:6]                    | I                 | N(17·16)   |             |           |           |  |  |  |  |
| MR[1]               |   |                                   |                                |                   |            |             |           |           |  |  |  |  |
| CONTROLO_L          |   |                                   |                                |                   |            |             |           |           |  |  |  |  |
| MB[1]               |   |                                   |                                |                   |            |             |           |           |  |  |  |  |
| LAFM0               | -2, 11 141  |                                   |                                | EXTID_LAFM[17:16] |            |             |           |           |  |  |  |  |
|                     | Mame  TCMR2  TTTSEL  MB[0]. CONTROLO_H  MB[0]. LAFM0  MB[0]. LAFM1  MB[0]. DATA_01  MB[0]. DATA_23  MB[0]. DATA_45  MB[0]. DATA_45  MB[0]. DATA_67  MB[0]. CONTROL1  MB[0]. CONTROL1  MB[0]. CONTROL1  MB[1]. CONTROLO_H  MB[1]. CONTROLO_L  MB[1]. | Name   31/23/15/7                 | Name   31/23/15/7   30/22/14/6 | Name              | Name       | Name        | Name      | Name      |  |  |  |  |

| Module   | Register<br>Name | 31/23/15/7 | 30/22/14/6                               | 29/21/13/5 | 28/20/12/4    | 27/19/11/3     | 26/18/10/2        | 25/17/9/1 | 24/16/8/0  |  |  |  |
|----------|------------------|------------|--|------------|---------------|----------------|-------------------|-----------|------------|--|--|--|
| RCAN-TL1 | MB[1].           |            | <u> </u>                                 | <u>I</u>   | EXTID_LA      | L<br>.FM[15:8] | <u>l</u>          |           |            |  |  |  |
| (RCAN_A) | LAFM1            |            |  |            | EXTID_L       | AFM[7:0]       |                   |           |            |  |  |  |
|          | MB[1].           |            |  |            | MSG_D         | ATA_0          |                   |           |            |  |  |  |
|          | DATA_01          |            |  |            | MSG_D         | ATA_1          |                   |           |            |  |  |  |
|          | MB[1].           |            |  |            | MSG_D         | ATA_2          |                   |           |            |  |  |  |
|          | DATA_23          |            |  |            | MSG_D         | ATA_3          |                   |           |            |  |  |  |
|          | MB[1].           |            |  |            | MSG_D         | ATA_4          |                   |           |            |  |  |  |
|          | DATA_45          |            |  |            | MSG_D         | ATA_5          |                   |           |            |  |  |  |
|          | MB[1].           |            |  |            | MSG_D         | ATA_6          |                   |           |            |  |  |  |
|          | DATA_67          |            |  |            | MSG_D         | ATA_7          |                   |           |            |  |  |  |
|          | MB[1].           | _          |  | NMC        | ATX           | DART           | MBC[2:0] DLC[3:0] |           |            |  |  |  |
|          | CONTROL1         |            | _  | _          |               |                |                   |           |            |  |  |  |
|          | MB[1].           |            | TimeStamp[15:8]                          |            |               |                |                   |           |            |  |  |  |
|          | TIMESTAMP        |            | TimeStamp[7:0]                           |            |               |                |                   |           |            |  |  |  |
|          | MB[2]            |            | Configured in the same order as in MB[1] |            |               |                |                   |           |            |  |  |  |
|          | :                |            | ÷  |            |               |                |                   |           |            |  |  |  |
|          | MB[15]           |            |  | Configure  | ed in the san | ne order as i  | n MB[1]           |           |            |  |  |  |
|          | MB[16].          | IDE        | RTR                                      | _          |               | :              | STDID[10:6]       |           |            |  |  |  |
|          | CONTROL0_H       |            |  | STDII      | D[5:0]        |                |                   | EXTID     | [17:16]    |  |  |  |
|          | MB[16].          |            |  |            | EXTID         | [15:8]         |                   |           |            |  |  |  |
|          | CONTROL0_L       |            |  |            | EXTIC         | 0[7:0]         |                   |           |            |  |  |  |
|          | MB[16].          | IDE_LAFM   |  | _          |               | STE            | DID_LAFM[1        | 0:6]      |            |  |  |  |
|          | LAFM0            |            |  | STDID_L    | AFM[5:0]      |                |                   | EXTID_LA  | AFM[17:16] |  |  |  |
|          | MB[16].          |            |  |            | EXTID_LA      | .FM[15:8]      |                   |           |            |  |  |  |
|          | LAFM1            |            |  |            | EXTID_L       | AFM[7:0]       |                   |           |            |  |  |  |
|          | MB[16].          |            |  |            | MSG_D         | ATA_0          |                   |           |            |  |  |  |
|          | DATA_01          |            |  |            | MSG_D         | ATA_1          |                   |           |            |  |  |  |
|          | MB[16].          |            |  |            | MSG_D         | ATA_2          |                   |           |            |  |  |  |
|          | DATA_23          | MSG_DATA_3 |  |            |               |                |                   |           |            |  |  |  |
|          | MB[16].          |            |  |            | MSG_D         | ATA_4          |                   |           |            |  |  |  |
|          | DATA_45          |            | MSG_DATA_5                               |            |               |                |                   |           |            |  |  |  |

Apr 01, 2014

| Module   | Register<br>Name | 31/23/15/7      | 30/22/14/6                 | 29/21/13/5 | 28/20/12/4    | 27/19/11/3     | 26/18/10/2  | 25/17/9/1  | 24/16/8/0  |  |  |  |
|----------|------------------|-----------------|----------------------------|------------|---------------|----------------|-------------|------------|------------|--|--|--|
| RCAN-TL1 | MB[16].          |                 |                            |            | MSG_D         | ATA_6          |             |            |            |  |  |  |
| (RCAN_A) | DATA_67          |                 |                            |            | MSG_D         | ATA_7          |             |            |            |  |  |  |
|          | MB[16].          | _               | _                          | NMC        | ATX           | DART           |             | MBC[2:0]   |            |  |  |  |
|          | CONTROL1         | _               | _                          |            | _             |                | DLC         | [3:0]      |            |  |  |  |
|          | MB[17]           |                 |                            | Configure  | ed in the sam | ne order as ir | n MB[16]    |            |            |  |  |  |
|          | :                |                 |                            |            | :             |                |             |            |            |  |  |  |
|          | MB[23]           |                 |                            | Configure  | ed in the sam | ne order as ir | n MB[16]    |            |            |  |  |  |
|          | MB[24].          | IDE             | RTR                        | _          |               | ;              | STDID[10:6] |            |            |  |  |  |
|          | CONTROL0_H       |                 |                            | STDII      | D[5:0]        |                |             | EXTID      | [17:16]    |  |  |  |
|          | MB[24].          |                 |                            |            | EXTID         | [15:8]         |             |            |            |  |  |  |
|          | CONTROL0_L       |                 |                            | EXTID[7:0] |               |                |             |            |            |  |  |  |
|          | MB[24]. I        | IDE_LAFM        | DE_LAFM — STDID_LAFM[10:6] |            |               |                |             |            |            |  |  |  |
|          | LAFM0            |                 |                            | STDID_L    | AFM[5:0]      |                |             | EXTID_LA   | AFM[17:16] |  |  |  |
|          | MB[24].          |                 | EXTID_LAFM[15:8]           |            |               |                |             |            |            |  |  |  |
|          | LAFM1            | EXTID_LAFM[7:0] |                            |            |               |                |             |            |            |  |  |  |
|          | MB[24].          |                 |                            |            | MSG_D         | ATA_0          |             |            |            |  |  |  |
|          | DATA_01          |                 |                            |            | MSG_D         | ATA_1          |             |            |            |  |  |  |
|          | MB[24].          |                 |                            |            | MSG_D         | ATA_2          |             |            |            |  |  |  |
|          | DATA_23          |                 |                            |            | MSG_D         | ATA_3          |             |            |            |  |  |  |
|          | MB[24].          |                 |                            |            | MSG_D         | ATA_4          |             |            |            |  |  |  |
|          | DATA_45          |                 |                            |            | MSG_D         | ATA_5          |             |            |            |  |  |  |
|          | MB[24].          |                 |                            |            | MSG_D         | ATA_6          |             |            |            |  |  |  |
|          | DATA_67          |                 |                            |            | MSG_D         | ATA_7          |             |            |            |  |  |  |
|          | MB[24].          | _               | _                          | NMC        | ATX           | DART           |             | MBC[2:0]   |            |  |  |  |
|          | CONTROL1         | _               | _                          | _          | _             |                | DLC         | [3:0]      |            |  |  |  |
|          | MB[24].          | тт              |                            |            |               |                |             |            |            |  |  |  |
|          | TTT              | тт              |                            |            |               |                |             |            |            |  |  |  |
|          | MB[24].          | TTW             | [1:0]                      |            |               | Off            | set         |            |            |  |  |  |
|          | TTCONTROL        | _               |                            | _          | _             | _              |             | Rep_Factor |            |  |  |  |

| Module   | Register<br>Name | 31/23/15/7            | 30/22/14/6                | 29/21/13/5       | 28/20/12/4    | 27/19/11/3    | 26/18/10/2  | 25/17/9/1 | 24/16/8/0 |  |  |  |
|----------|------------------|-----------------------|---------------------------|------------------|---------------|---------------|-------------|-----------|-----------|--|--|--|
| RCAN-TL1 | MB[25]           |                       | <u>I</u>                  | Configure        | ed in the sam | ne order as i | n MB[24]    |           | 1         |  |  |  |
| (RCAN_A) | :                |                       |                           |                  | :             |               |             |           |           |  |  |  |
|          | MB[29]           |                       |                           | Configure        | ed in the sam | ne order as i | n MB[24]    |           |           |  |  |  |
|          | MB[30].          | IDE                   | RTR                       | _                |               | :             | STDID[10:6] |           |           |  |  |  |
|          | CONTROL0_H       |                       | •                         | STDI             | D[5:0]        |               |             | EXTID     | [17:16]   |  |  |  |
|          | MB[30].          |                       |                           |                  | EXTID         | [15:8]        |             | •         |           |  |  |  |
|          | CONTROL0_L       |                       |                           |                  | EXTI          | D[7:0]        |             |           |           |  |  |  |
|          | MB[30].          | ·                     |                           |                  |               |               |             |           |           |  |  |  |
|          | LAFM0            |                       | STDID_LAFM[5:0] EXTID_LAF |                  |               |               |             |           |           |  |  |  |
|          | MB[30].          |                       |                           | EXTID_LAFM[15:8] |               |               |             |           |           |  |  |  |
|          | LAFM1            | EXTID_LAFM[7:0]       |                           |                  |               |               |             |           |           |  |  |  |
|          | MB[30].          |                       |                           |                  | MSG_D         | ATA_0         |             |           |           |  |  |  |
|          | DATA_01          |                       |                           |                  | MSG_D         | ATA_1         |             |           |           |  |  |  |
|          | MB[30].          |                       | MSG_DATA_2 MSG_DATA_3     |                  |               |               |             |           |           |  |  |  |
|          | DATA_23          |                       |                           |                  |               |               |             |           |           |  |  |  |
|          | MB[30].          |                       |                           |                  | MSG_D         | ATA_4         |             |           |           |  |  |  |
|          | DATA_45          |                       |                           |                  | MSG_D         | ATA_5         |             |           |           |  |  |  |
|          | MB[30].          |                       |                           |                  | MSG_D         | ATA_6         |             |           |           |  |  |  |
|          | DATA_67          |                       |                           |                  | MSG_D         | ATA_7         |             |           |           |  |  |  |
|          | MB[30].          | _                     | _                         | NMC              | ATX           | DART          |             | MBC[2:0]  |           |  |  |  |
|          | CONTROL1         | _                     | _                         | _                | _             |               | DLC         | [3:0]     |           |  |  |  |
|          | MB[30].          |                       |                           |                  | TimeStar      | mp[15:8]      |             |           |           |  |  |  |
|          | TIMESTAMP        |                       |                           |                  | TimeSta       | mp[7:0]       |             |           |           |  |  |  |
|          | MB[30].          |                       |                           |                  | TT            | Т             |             |           |           |  |  |  |
|          | TTT              |                       |                           |                  | тт            | Т             |             |           |           |  |  |  |
|          | MB[31].          | IDE RTR — STDID[10:6] |                           |                  |               |               |             |           |           |  |  |  |
|          | CONTROL0_H       |                       |                           | STDI             | D[5:0]        |               |             | EXTID     | [17:16]   |  |  |  |
|          | MB[31].          |                       |                           |                  | EXTID         | [15:8]        |             |           |           |  |  |  |
|          | CONTROL0_L       |                       |                           |                  | EXTI          | 0[7:0]        |             |           |           |  |  |  |

| Module               | Register<br>Name    | 31/23/15/7 | 30/22/14/6      | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2  | 25/17/9/1 | 24/16/8/0  |  |  |  |  |
|----------------------|---------------------|------------|-----------------|------------|------------|------------|-------------|-----------|------------|--|--|--|--|
| RCAN-TL1             | MB[31].             | IDE_LAFM   |                 |            |            | STE        | DID_LAFM[10 | 0:6]      |            |  |  |  |  |
| (RCAN_A)             | LAFM0               |            |                 | STDID_L    | AFM[5:0]   |            |             | EXTID_LA  | .FM[17:16] |  |  |  |  |
|                      | MB[31].             |            |                 |            | EXTID_LA   | AFM[15:8]  |             |           |            |  |  |  |  |
|                      | LAFM1               |            | EXTID_LAFM[7:0] |            |            |            |             |           |            |  |  |  |  |
|                      | MB[31].             | MSG_DATA_0 |                 |            |            |            |             |           |            |  |  |  |  |
|                      | DATA_01             |            | MSG_DATA_1      |            |            |            |             |           |            |  |  |  |  |
|                      | MB[31].             |            |                 |            | MSG_D      | ATA_2      |             |           |            |  |  |  |  |
|                      | DATA_23             |            |                 |            | MSG_D      | ATA_3      |             |           |            |  |  |  |  |
|                      | MB[31].             |            |                 |            |            |            |             |           |            |  |  |  |  |
|                      | DATA_45             |            | MSG_DATA_5      |            |            |            |             |           |            |  |  |  |  |
|                      | MB[31].             |            |                 |            | MSG_D      | ATA_6      |             |           |            |  |  |  |  |
|                      | DATA_67             |            |                 |            | MSG_D      | ATA_7      |             |           |            |  |  |  |  |
|                      | MB[31].<br>CONTROL1 |            | _               | NMC        | ATX        | DART       |             | MBC[2:0]  |            |  |  |  |  |
|                      |                     |            |                 |            |            |            |             |           |            |  |  |  |  |
|                      | MB[31].             |            | TimeStamp[15:8] |            |            |            |             |           |            |  |  |  |  |
|                      | TIMESTAMP           |            | 1               | ı          | TimeSta    | imp[7:0]   | 1           | 1         | 1          |  |  |  |  |
|                      | MBESR               | _          | _               | _          | _          | _          | _           | _         | _          |  |  |  |  |
|                      |                     | _          | _               | _          | _          | _          | _           | _         | MBEF       |  |  |  |  |
|                      | MBECR               |            | _               | _          | _          | _          | _           | _         | _          |  |  |  |  |
|                      |                     |            |                 |            |            |            |             |           | MBIM       |  |  |  |  |
| RCAN-TL1<br>(RCAN_B) | MCR                 | MCR15      | MCR14           |            |            |            |             | TST[2:0]  | 1          |  |  |  |  |
| (NCAN_B)             |                     | MCR7       | MCR6            | MCR5       |            |            | MCR2        | MCR1      | MCR0       |  |  |  |  |
|                      | GSR                 |            |                 | _          |            |            |             |           | _          |  |  |  |  |
|                      |                     |            |                 | GSR5       | GSR4       | GSR3       | GSR2        | GSR1      | GSR0       |  |  |  |  |
|                      | BCR1                |            | TSG             | 1[3:0]     |            |            |             | TSG2[2:0] |            |  |  |  |  |
|                      |                     |            |                 | SJW        | /[1:0]     |            |             |           | BSP        |  |  |  |  |
|                      | BCR0                |            | _               | _          |            | _          |             | _         |            |  |  |  |  |
|                      |                     | BRP[7:0]   |                 |            |            |            |             |           |            |  |  |  |  |
|                      | IRR                 | IRR15      | IRR14           | IRR13      | IRR12      | IRR11      | IRR10       | IRR9      | IRR8       |  |  |  |  |
|                      |                     | IRR7       | IRR6            | IRR5       | IRR4       | IRR3       | IRR2        | IRR1      | IRR0       |  |  |  |  |

| Module   | Register<br>Name | 31/23/15/7   | 30/22/14/6  | 29/21/13/5 | 28/20/12/4        | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |  |
|----------|------------------|--------------|-------------|------------|-------------------|------------|------------|-----------|-----------|--|--|--|--|--|
| RCAN-TL1 | IMR              | IMR15        | IMR14       | IMR13      | IMR12             | IMR11      | IMR10      | IMR9      | IMR8      |  |  |  |  |  |
| (RCAN_B) |                  | 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            |              | TXPR1[15:8] |            |                   |            |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | TXPR              | 1[7:0]     |            |           |           |  |  |  |  |  |
|          | TXPR0            |              |             |            | TXPR              | )[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | TXPR0[7:1]        |            |            |           |           |  |  |  |  |  |
|          | TXCR1            |              |             |            | TXCR              | 1[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |              | TXCR1[7:0]  |            |                   |            |            |           |           |  |  |  |  |  |
|          | TXCR0            | TXCR0[15:8]  |             |            |                   |            |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | TXCR0[7:1]        |            |            |           |           |  |  |  |  |  |
|          | TXACK1           | TXACK1[15:8] |             |            |                   |            |            |           |           |  |  |  |  |  |
|          |                  |              | TXACK1[7:0] |            |                   |            |            |           |           |  |  |  |  |  |
|          | TXACK0           |              |             |            | TXACK             | 0[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | TXACK0[7:1        | ]          |            |           | _         |  |  |  |  |  |
|          | ABACK1           |              |             |            | ABACK             | 1[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | ABAC              | K1[7:0]    |            |           |           |  |  |  |  |  |
|          | ABACK0           |              |             |            | ABACK             | 0[15:8]    |            |           | _         |  |  |  |  |  |
|          |                  |              |             |            | ABACK0[7:1        | ]          |            |           | _         |  |  |  |  |  |
|          | RXPR1            |              |             |            | RXPR <sup>-</sup> | 1[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | RXPR              | 1[7:0]     |            |           |           |  |  |  |  |  |
|          | RXPR0            |              |             |            | RXPR              | 0[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | RXPR              | 0[7:0]     |            |           |           |  |  |  |  |  |
|          | RFPR1            |              |             |            | RFPR <sup>-</sup> | 1[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | RFPR              | 1[7:0]     |            |           |           |  |  |  |  |  |
|          | RFPR0            |              |             |            | RFPR              | )[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | RFPR              | 0[7:0]     |            |           |           |  |  |  |  |  |
|          | MBIMR1           |              |             |            | MBIMR             | 1[15:8]    |            |           |           |  |  |  |  |  |
|          |                  |              |             |            | MBIMF             | R1[7:0]    |            |           |           |  |  |  |  |  |

Apr 01, 2014

| Module   | Register<br>Name | 31/23/15/7             | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |
|----------|------------------|------------------------|------------|------------|------------|------------|------------|-----------|-----------|--|--|
| RCAN-TL1 | MBIMR0           |                        |            | I          | MBIMR      | 0[15:8]    | I          | I         | ı.        |  |  |
| (RCAN_B) |                  |                        |            |            | MBIME      | R0[7:0]    |            |           |           |  |  |
|          | UMSR1            |                        |            |            | UMSR       | 1[15:8]    |            |           |           |  |  |
|          |                  |                        |            |            | UMSF       | R1[7:0]    |            |           |           |  |  |
|          | UMSR0            |                        |            |            | UMSR       | 0[15:8]    |            |           |           |  |  |
|          |                  |                        |            |            | UMSF       | R0[7:0]    |            |           |           |  |  |
|          | TTCR0            | TCR15                  | TCR14      | TCR13      | TCR12      | TCR11      | TCR10      | _         | _         |  |  |
|          |                  | _                      | TCR6       | TPSC5      | TPSC4      | TPSC3      | TPSC2      | TPSC1     | TPSC0     |  |  |
|          | CMAX_TEW         |                        |            |            | _          | _          |            | CMAX[2:0] |           |  |  |
|          |                  | _                      | _          |            | _          |            | TEW[3:0]   |           |           |  |  |
|          | RFTROFF          |                        | r          | 1          | RFTRO      | FF[7:0]    | 1          | 1         |           |  |  |
|          |                  | _                      | _          | _          | _          | _          | _          | _         | _         |  |  |
|          | TSR              | _                      | _          | _          | _          |            | _          | _         | _         |  |  |
|          |                  | _                      |            | _          | TSR4       | TSR3       | TSR2       | TSR1      | TSR0      |  |  |
|          | CCR              |                        |            |            |            |            |            |           |           |  |  |
|          |                  |                        |            |            |            | CCR        | [5:0]      |           |           |  |  |
|          | TCNTR            |                        |            |            | TCNTF      |            |            |           |           |  |  |
|          |                  |                        |            |            | TCNT       |            |            |           |           |  |  |
|          | CYCTR            |                        |            |            | СҮСТЕ      | -          |            |           |           |  |  |
|          |                  |                        |            |            | CYCT       |            |            |           |           |  |  |
|          | RFMK             |                        |            |            | RFMK       |            |            |           |           |  |  |
|          | TOLIDA           |                        |            |            | RFM        |            |            |           |           |  |  |
|          | TCMR0            |                        |            |            | TCMR       |            |            |           |           |  |  |
|          | TCMR1            |                        |            |            | TCMR       |            |            |           |           |  |  |
|          | TOWN             |                        |            |            | TCMR       |            |            |           |           |  |  |
|          | TCMR2            |                        |            |            | TCMR       |            |            |           |           |  |  |
|          | TOWING           | TCMR2[15:8] TCMR2[7:0] |            |            |            |            |            |           |           |  |  |
|          | TTTSEL           |                        |            |            |            |            | 1          |           |           |  |  |
|          | ITTOLL           | — TTTSEL[14:8]         |            |            |            |            |            |           |           |  |  |
|          |                  |                        |            |            |            |            |            |           |           |  |  |

| Module   | Register<br>Name | 31/23/15/7 | 30/22/14/6    | 29/21/13/5    | 28/20/12/4        | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |
|----------|------------------|------------|---------------|---------------|-------------------|------------|------------|-----------|-----------|--|
| RCAN-TL1 | MB[0]            | Configured | I in the same | order as in I | MB[0] of the      | RCAN_A.    | I          | <u>I</u>  |           |  |
| (RCAN_B) | MB[1]            | Configured | I in the same | order as in I | MB[1] of the      | RCAN_A.    |            |           |           |  |
|          | MB[2]            | Configured | I in the same | order as in I | MB[2] of the      | RCAN_A.    |            |           |           |  |
|          | :                |            |               |               | :                 |            |            |           |           |  |
|          | MB[29]           | Configured | I in the same | order as in I | MB[29] of the     | RCAN_A.    |            |           |           |  |
|          | MB[30]           | Configured | I in the same | order as in I | MB[30] of the     | e RCAN_A.  |            |           |           |  |
|          | MB[31]           | Configured | I in the same | order as in I | MB[31] of the     | RCAN_A.    |            |           |           |  |
|          | MBESR            | _          |               | _             |                   | _          |            | _         |           |  |
|          |                  | _          | _             | _             | _                 | _          | _          | _         | MBEF      |  |
|          | MBECR            | _          | _             | _             | _                 | _          | _          | _         | _         |  |
|          |                  | _          | _             | _             | _                 | _          | _          | _         | МВІМ      |  |
| RCAN-TL1 | MCR              | MCR15      | MCR14         | _             | _                 | _          |            | TST[2:0]  |           |  |
| (RCAN_C) |                  | MCR7       | MCR6          | MCR5          |                   | _          | MCR2       | MCR1      | MCR0      |  |
|          | GSR              | _          | _             | _             | _                 | _          |            | _         | _         |  |
|          |                  |            | _             | GSR5          | GSR4              | GSR3       | GSR2       | GSR1      | GSR0      |  |
|          | BCR1             |            | TSG           | 1[3:0]        |                   | _          |            | TSG2[2:0] |           |  |
|          |                  | _          | _             | SJW[1:0]      |                   | _          | _          | _         | BSP       |  |
|          | BCR0             | _          | _             |               |                   |            |            |           |           |  |
|          |                  |            |               |               | BRP               | [7:0]      |            |           |           |  |
|          | IRR              | IRR15      | IRR14         | IRR13         | IRR12             | IRR11      | IRR10      | IRR9      | IRR8      |  |
|          |                  | IRR7       | IRR6          | IRR5          | IRR4              | IRR3       | IRR2       | IRR1      | IRR0      |  |
|          | IMR              | IMR15      | IMR14         | IMR13         | IMR12             | IMR11      | IMR10      | IMR9      | IMR8      |  |
|          |                  | IMR7       | IMR6          | IMR5          | IMR4              | IMR3       | IMR2       | IMR1      | IMR0      |  |
|          | TEC/REC          | TEC7       | TEC6          | TEC5          | TEC4              | TEC3       | TEC2       | TEC1      | TEC0      |  |
|          |                  | REC7       | REC6          | REC5          | REC4              | REC3       | REC2       | REC1      | REC0      |  |
|          | TXPR1            |            |               |               | TXPR <sup>-</sup> | 1[15:8]    |            |           |           |  |
|          |                  |            |               |               | TXPR              | 1[7:0]     |            |           |           |  |
|          | TXPR0            |            |               |               | TXPR              | 0[15:8]    |            |           | _         |  |
|          |                  |            |               |               |                   |            |            |           |           |  |
|          | TXCR1            |            |               |               | TXCR              | 1[15:8]    |            |           |           |  |
|          |                  |            |               |               | TXCR              | 1[7:0]     |            |           |           |  |

| Module   | Register<br>Name | 31/23/15/7    | 30/22/14/6  | 29/21/13/5 | 28/20/12/4        | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |  |  |  |
|----------|------------------|---------------|-------------|------------|-------------------|------------|------------|-----------|-----------|--|--|--|--|
| RCAN-TL1 | TXCR0            |               |             | ı          | TXCR              | D[15:8]    | ı          | <u>I</u>  | ·         |  |  |  |  |
| (RCAN_C) |                  |               |             |            | TXCR0[7:1]        |            |            |           |           |  |  |  |  |
|          | TXACK1           |               |             |            | TXACK             | 1[15:8]    |            |           |           |  |  |  |  |
|          |                  |               |             |            | TXAC              | K1[7:0]    |            |           |           |  |  |  |  |
|          | TXACK0           |               |             |            | TXACK             | 0[15:8]    |            |           |           |  |  |  |  |
|          |                  |               |             |            | TXACK0[7:1        | ]          |            |           | _         |  |  |  |  |
|          | ABACK1           |               |             |            | ABACK             | [1[15:8]   |            |           |           |  |  |  |  |
|          |                  |               |             |            | ABAC              | K1[7:0]    |            |           |           |  |  |  |  |
|          | ABACK0           |               |             |            | ABACK             | 0[15:8]    |            |           |           |  |  |  |  |
|          |                  |               |             |            | ABACK0[7:1        | ]          |            |           | _         |  |  |  |  |
|          | RXPR1            |               | RXPR1[15:8] |            |                   |            |            |           |           |  |  |  |  |
|          |                  |               |             |            | RXPR              | 1[7:0]     |            |           |           |  |  |  |  |
|          | RXPR0            |               |             |            |                   |            |            |           |           |  |  |  |  |
|          |                  |               |             |            | RXPR              | 0[7:0]     |            |           |           |  |  |  |  |
|          | RFPR1            |               |             |            | RFPR <sup>-</sup> | 1[15:8]    |            |           |           |  |  |  |  |
|          |                  |               |             |            | RFPR              |            |            |           |           |  |  |  |  |
|          | RFPR0            |               |             |            | RFPR              |            |            |           |           |  |  |  |  |
|          |                  |               |             |            | RFPR              |            |            |           |           |  |  |  |  |
|          | MBIMR1           |               |             |            | MBIMR             |            |            |           |           |  |  |  |  |
|          |                  |               |             |            | MBIMF             |            |            |           |           |  |  |  |  |
|          | MBIMR0           |               |             |            | MBIMR             |            |            |           |           |  |  |  |  |
|          | LIN AGE A        |               |             |            | MBIMF             |            |            |           |           |  |  |  |  |
|          | UMSR1            |               |             |            | UMSR              |            |            |           |           |  |  |  |  |
|          | LIMODO           |               |             |            | UMSR              |            |            |           |           |  |  |  |  |
|          | UMSR0            |               |             |            | UMSR              |            |            |           |           |  |  |  |  |
|          | TTCR0            | TOP15         | TCR14       | TCR13      | UMSR<br>TCR12     | TCR11      | TCR10      |           |           |  |  |  |  |
|          | ITCHU            | TCR15         |             |            |                   |            |            |           | TDCCC     |  |  |  |  |
|          | CMAY TEM         | <del> -</del> | TCR6        | TPSC5      | TPSC4             | TPSC3      | TPSC2      | TPSC1     | TPSC0     |  |  |  |  |
|          | CMAX_TEW         |               |             |            |                   |            |            | CMAX[2:0] |           |  |  |  |  |
|          | TEW[3:0]         |               |             |            |                   |            |            |           |           |  |  |  |  |

| Module   | Register<br>Name | 31/23/15/7 | 30/22/14/6  | 29/21/13/5    | 28/20/12/4 | 27/19/11/3   | 26/18/10/2  | 25/17/9/1 | 24/16/8/0 |  |  |  |  |  |
|----------|------------------|------------|-------------|---------------|------------|--------------|-------------|-----------|-----------|--|--|--|--|--|
| RCAN-TL1 | RFTROFF          |            |             |               | RFTRO      | FF[7:0]      |             |           | ,         |  |  |  |  |  |
| (RCAN_C) |                  |            |             | _             |            |              | _           | _         |           |  |  |  |  |  |
|          | TSR              | _          |             | _             | _          | _            | _           | _         | _         |  |  |  |  |  |
|          |                  |            | _           |               | TSR4       | TSR3         | TSR2        | TSR1      | TSR0      |  |  |  |  |  |
|          | CCR              | _          |             |               |            |              |             |           |           |  |  |  |  |  |
|          |                  | _          |             |               |            | CCR          | [5:0]       |           |           |  |  |  |  |  |
|          | TCNTR            |            |             |               | TCNTF      | R[15:8]      |             |           |           |  |  |  |  |  |
|          |                  |            |             |               | TCNT       | R[7:0]       |             |           |           |  |  |  |  |  |
|          | CYCTR            |            |             |               | CYCTF      | R[15:8]      |             |           |           |  |  |  |  |  |
|          |                  |            | CYCTR[7:0]  |               |            |              |             |           |           |  |  |  |  |  |
|          | RFMK             |            | RFMK[15:8]  |               |            |              |             |           |           |  |  |  |  |  |
|          |                  |            |             |               | RFM        | K[7:0]       |             |           |           |  |  |  |  |  |
|          | TCMR0            |            |             |               | TCMR       | 0[15:8]      |             |           |           |  |  |  |  |  |
|          |                  |            | TCMR0[7:0]  |               |            |              |             |           |           |  |  |  |  |  |
|          | TCMR1            |            | TCMR1[15:8] |               |            |              |             |           |           |  |  |  |  |  |
|          |                  |            |             |               | TCMR       | 1[7:0]       |             |           |           |  |  |  |  |  |
|          | TCMR2            |            |             |               | TCMR       | 2[15:8]      |             |           |           |  |  |  |  |  |
|          |                  |            | T           |               | TCMR       | 2[7:0]       |             |           |           |  |  |  |  |  |
|          | TTTSEL           | _          |             | 1             | Т.         | TTSEL[14:8   | ]           | 1         | ,         |  |  |  |  |  |
|          |                  | _          | _           | _             | _          |              | _           | _         | _         |  |  |  |  |  |
|          | MB[0]            |            | Conf        | igured in the | same order | as in MB[0]  | of the RCAN | I_A.      |           |  |  |  |  |  |
|          | MB[1]            |            | Conf        | igured in the | same order | as in MB[1]  | of the RCAN | I_A.      |           |  |  |  |  |  |
|          | MB[2]            |            | Conf        | igured in the | same order | as in MB[2]  | of the RCAN | I_A.      |           |  |  |  |  |  |
|          | :                |            |             |               | :          |              |             |           |           |  |  |  |  |  |
|          | MB[29]           |            | Confi       | gured in the  | same order | as in MB[29] | of the RCA  | N_A.      |           |  |  |  |  |  |
|          | MB[30]           |            | Confi       | gured in the  | same order | as in MB[30] | of the RCA  | N_A.      |           |  |  |  |  |  |
|          | MB[31]           |            | Confi       | gured in the  | same order | as in MB[31] | of the RCA  | N_A.      | ,         |  |  |  |  |  |
|          | MBESR            | _          |             | _             | _          | _            | _           | _         | _         |  |  |  |  |  |
|          |                  |            |             | _             | _          |              | _           | _         | MBEF      |  |  |  |  |  |
|          | MBECR            |            | _           | _             | _          | _            | _           | _         | _         |  |  |  |  |  |
|          |                  | _          | _           | _             | _          | _            | _           | _         | MBIM      |  |  |  |  |  |

| Module      | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|-------------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| ADC (ADC_A) | ADRD0            |            |            |            |            |            |            |           |           |
|             | ADR0             |            |            |            |            |            |            |           |           |
|             | ADR1             |            |            |            |            |            |            |           |           |
|             | ADR2             |            |            |            |            |            |            |           |           |
|             | ADR3             |            |            |            |            |            |            |           |           |
|             | ADR4             |            |            |            |            |            |            |           |           |
|             | ADR5             |            |            |            |            |            |            |           |           |
|             | ADR6             |            |            |            |            |            |            |           |           |
|             | ADR7             |            |            |            |            |            |            |           |           |
|             | ADR8             |            |            |            |            |            |            |           |           |
|             | ADR9             |            |            |            |            |            |            |           |           |
|             | ADR10            |            |            |            |            |            |            |           |           |
|             | ADR11            |            |            |            |            |            |            |           |           |
|             | ADR12            |            |            |            |            |            |            |           |           |
|             | ADR13            |            |            |            |            |            |            |           |           |
|             |                  |            |            |            |            |            |            |           |           |

|             | Register |            |            |            |            |            |            |           |           |
|-------------|----------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| Module      | Name     | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
| ADC (ADC_A) | ADR14    |            |            |            |            |            |            |           |           |
|             | ADR15    |            |            |            |            |            |            |           |           |
|             |          |            |            |            |            |            |            |           |           |
|             | ADR16    |            |            |            |            |            |            |           |           |
|             | ADR17    |            |            |            |            |            |            |           |           |
|             | ADR18    |            |            |            |            |            |            |           |           |
|             |          |            |            |            |            |            |            |           |           |
|             | ADR19    |            |            |            |            |            |            |           |           |
|             | ADR20    |            |            |            |            |            |            |           |           |
|             |          |            |            |            |            |            |            |           |           |
|             | ADR21    |            |            |            |            |            |            |           |           |
|             | ADR22    |            |            |            |            |            |            |           |           |
|             |          |            |            |            |            |            |            |           |           |
|             | ADR23    |            |            |            |            |            |            |           |           |
|             | ADR24    |            |            |            |            |            |            |           |           |
|             |          |            |            |            |            |            |            |           |           |
|             | ADR25    |            |            |            |            |            |            |           |           |
|             | ADR26    |            |            |            |            |            |            |           |           |
|             | ADR27    |            |            |            |            |            |            |           |           |
|             | AUNZI    |            |            |            |            |            |            |           |           |
| ADC (ADC_B) | ADRD1    |            |            |            |            |            |            |           |           |
|             |          |            |            |            |            |            |            |           |           |

RENESAS

| Module      | Register | 24/22/45/7 | 20/22/4 4/6 | 20/24/42/5 | 28/20/12/4 | 27/10/11/2 | 26/49/40/2 | 25/47/0/4 | 24/16/8/0 |
|-------------|----------|------------|-------------|------------|------------|------------|------------|-----------|-----------|
|             | Name     | 31/23/15// | 30/22/14/6  | 29/21/13/5 | 20/20/12/4 | 21/19/11/3 | 20/10/10/2 | 25/17/9/1 | 24/10/0/0 |
| ADC (ADC_B) | ADR40    |            |             |            |            |            |            |           |           |
|             | ADR41    |            |             |            |            |            |            |           |           |
|             |          |            |             |            |            |            |            |           |           |
|             | ADR42    |            |             |            |            |            |            |           |           |
|             |          |            |             |            |            |            |            |           |           |
|             | ADR43    |            |             |            |            |            |            |           |           |
|             | ADR44    |            |             |            |            |            |            |           |           |
|             |          |            |             |            |            |            |            |           |           |
|             | ADR45    |            |             |            |            |            |            |           |           |
|             | ADR46    |            |             |            |            |            |            |           |           |
|             |          |            |             |            |            |            |            |           |           |
|             | ADR47    |            |             |            |            |            |            |           |           |
|             |          |            |             |            |            |            |            |           |           |
|             | ADR48    |            |             |            |            |            |            |           |           |
| ADC (ADC_A) | ADCSR0   | ADST       | ADCS        | _          | ADIE       | _          | _          | TRGE      | EXTRG     |
| ADC (ADC_B) | ADCSR1   | ADST       | ADCS        | _          | ADIE       | _          | _          | TRGE      | EXTRG     |
| ADC (ADC_A) | ADREF0   | ADSCACT    | ADITACT     |            |            |            |            | _         | ADF       |
| ADC (ADC_B) | ADREF1   | ADSCACT    | ADITACT     | _          | _          | _          | _          |           | ADF       |
| ADC (ADC_A) | ADTRE0   | ADTRGE15   | ADTRGE14    | ADTRGE13   | ADTRGE12   | ADTRGE11   | ADTRGE10   | ADTRGE9   | ADTRGE8   |
|             |          | ADTRGE7    | ADTRGE6     | ADTRGE5    | ADTRGE4    | ADTRGE3    | ADTRGE2    | ADTRGE1   | ADTRGE0   |
| ADC (ADC_B) |          | ADTRGE47   | ADTRGE46    | ADTRGE45   | ADTRGE44   | ADTRGE43   | ADTRGE42   | ADTRGE41  | ADTRGE40  |
| ADC (ADC_A) | ADTRF0   | ADTF15     | ADTF14      | ADTF13     | ADTF12     | ADTF11     | ADTF10     | ADTF9     | ADTF8     |
|             |          | ADTF7      | ADTF6       | ADTF5      | ADTF4      | ADTF3      | ADTF2      | ADTF1     | ADTF0     |
| ADC (ADC_B) |          | ADTF47     | ADTF46      | ADTF45     | ADTF44     | ADTF43     | ADTF42     | ADTF41    | ADTF40    |
| ADC (ADC_A) | ADTRS0   | ADTRS15    | ADTRS14     | ADTRS13    | ADTRS12    | ADTRS11    | ADTRS10    | ADTRS9    | ADTRS8    |
|             |          | ADTRS7     | ADTRS6      | ADTRS5     | ADTRS4     | ADTRS3     | ADTRS2     | ADTRS1    | ADTRS0    |
| ADC (ADC_B) | ADTRS1   | ADTRS47    | ADTRS46     | ADTRS45    | ADTRS44    | ADTRS43    | ADTRS42    | ADTRS41   | ADTRS40   |

| Module      | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|-------------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| ADC (ADC_A) | ADSTRG0          | ADSTRG15   | ADSTRG14   | ADSTRG13   | ADSTRG12   | ADSTRG11   | ADSTRG10   | ADSTRG9   | ADSTRG8   |
|             |                  | ADSTRG7    | ADSTRG6    | ADSTRG5    | ADSTRG4    | ADSTRG3    | ADSTRG2    | ADSTRG1   | ADSTRG0   |
| ADC (ADC_B) | ADSTRG1          | ADSTRG47   | ADSTRG46   | ADSTRG45   | ADSTRG44   | ADSTRG43   | ADSTRG42   | ADSTRG41  | ADSTRG40  |
| ADC (ADC_A) | ADTRD0           | ADIDE15    | ADIDE14    | ADIDE13    | ADIDE12    | ADIDE11    | ADIDE10    | ADIDE9    | ADIDE8    |
|             |                  | ADIDE7     | ADIDE6     | ADIDE5     | ADIDE4     | ADIDE3     | ADIDE2     | ADIDE1    | ADIDE0    |
| ADC (ADC_B) | ADTRD1           | ADIDE47    | ADIDE46    | ADIDE45    | ADIDE44    | ADIDE43    | ADIDE42    | ADIDE41   | ADIDE40   |
| ADC (ADC_A) | ADADS0           | ADS7       | ADS6       | ADS5       | ADS4       | ADS3       | ADS2       | ADS1      | ADS0      |
| ADC (ADC_B) | ADADS1           | ADS47      | ADS46      | ADS45      | ADS44      | ADS43      | ADS42      | ADS41     | ADS40     |
| ADC (ADC_A) | ADADC0           | _          | _          | _          | _          | _          |            | ADC       | [1:0]     |
| ADC (ADC_B) | ADADC1           | _          | _          | _          | _          | _          | _          | ADC       | [1:0]     |
| ADC (ADC_A) | ADANS0           | ANS15      | ANS14      | ANS13      | ANS12      | ANS11      | ANS10      | ANS9      | ANS8      |
|             |                  | ANS7       | ANS6       | ANS5       | ANS4       | ANS3       | ANS2       | ANS1      | ANS0      |
|             | ADANS1           | _          | _          | _          | _          | ANS27      | ANS26      | ANS25     | ANS24     |
|             |                  | ANS23      | ANS22      | ANS21      | ANS20      | ANS19      | ANS18      | ANS17     | ANS16     |
| ADC (ADC_B) | ADANS3           |            | _          |            | _          |            |            |           | ANS48     |
|             |                  | ANS47      | ANS46      | ANS45      | ANS44      | ANS43      | ANS42      | ANS41     | ANS40     |
| ADC (ADC_A) | ADCER0           | ADRFMT     | _          | _          | _          | DIAGM      | DIAGLD     | DIAGV     | AL[1:0]   |
|             |                  | CKS        | _          | _          | _          |            |            |           | ITTRGS    |
| ADC (ADC_B) | ADCER1           | ADRFMT     |            | _          |            | DIAGM      | DIAGLD     | DIAGV     | AL[1:0]   |
|             |                  | CKS        | _          | _          | _          |            |            |           |           |
| JTAG        | SDIR             |            |            |            |            |            |            |           |           |
|             |                  |            |            |            |            |            |            |           |           |
|             | SDID             |            |            |            |            |            |            |           |           |
|             |                  |            |            |            |            |            |            |           |           |
|             | SDBPR            |            |            |            |            |            |            |           |           |
|             |                  |            |            |            |            |            |            |           |           |
|             | SDBSR            |            |            |            |            |            |            |           |           |
|             |                  |            |            |            |            |            |            |           |           |

| Module | Register<br>Name | 31/23/15/7         | 30/22/14/6 | 29/21/13/5           | 28/20/12/4 | 27/19/11/3      | 26/18/10/2      | 25/17/9/1      | 24/16/8/0 |
|--------|------------------|--------------------|------------|----------------------|------------|-----------------|-----------------|----------------|-----------|
| AUD-II | AUCSR            | CLK                | [1:0]      | BW                   | [1:0]      | ОС              | [1:0]           | BR[            | 1:0]      |
|        |                  | WA                 | [1:0]      | WB                   | [1:0]      | _               | ТМ              | _              | EN        |
|        | AUWASR           |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        | AUWAER           |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        | AUWBSR           |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        | AUWBER           |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        |                  |                    |            |                      |            |                 |                 |                |           |
|        | AU 5000          |                    |            |                      |            |                 |                 |                | 2.0       |
|        | AUECSR           | —                  | _          | - WDODIO.01          | _          | TDEV            |                 | WAOI           | 3[2:1]    |
| PFC    | PAIOR            | WA0B[0]<br>PA15IOR | PA14IOR    | WB0B[2:0]<br>PA13IOR | PA12IOR    | TREX<br>PA11IOR | TRSB<br>PA10IOR | TRGN<br>PA9IOR | PA8IOR    |
| FFC    | PAION            | PA7IOR             | PA6IOR     | PA5IOR               | PA4IOR     | PASIOR          | PA2IOR          | PATIOR         | PAOIOR    |
|        | PACR4            |                    | FAGION     | FASION               | PA15MD     | FASION          | FAZION          |                | PA14MD    |
|        | TAOTE            |                    |            |                      | PA13MD     |                 |                 |                | PA12MD    |
|        | PACR3            |                    |            |                      | PA11MD     |                 |                 |                | PA10MD    |
|        |                  |                    |            |                      | PA9MD      |                 |                 |                | PA8MD     |
|        | PACR2            |                    | _          | _                    | PA7MD      | _               | _               | _              | PA6MD     |
|        |                  |                    |            |                      | PA5MD      |                 |                 | _              | PA4MD     |
|        | PACR1            | _                  |            |                      | PA3MD      |                 |                 | _              | PA2MD     |
|        |                  | _                  |            |                      | PA1MD      |                 |                 |                | PA0MD     |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|------------|-----------|
| PFC    | PBIOR            | _          | PB14IOR    | PB13IOR    | PB12IOR    | PB11IOR    | PB10IOR    | PB9IOR     | PB8IOR    |
|        |                  | PB7IOR     | PB6IOR     | PB5IOR     | PB4IOR     | PB3IOR     | PB2IOR     | PB1IOR     | PB0IOR    |
|        | PBCR4            |            |            |            |            |            | F          | PB14MD[2:0 | ]         |
|        |                  |            | _          | PB13N      | /ID[1:0]   | _          |            | PB12N      | /ID[1:0]  |
|        | PBCR3            | _          | _          | PB11N      | ИD[1:0]    | _          |            | _          | PB10MD    |
|        |                  |            | _          | _          | PB9MD      | _          | _          | PB8M       | D[1:0]    |
|        | PBCR2            | _          | _          | _          | PB7MD      | _          |            | _          | PB6MD     |
|        |                  |            |            | PB5MD[2:0] |            | _          |            | PB4MD[2:0] |           |
|        | PBCR1            | _          | _          | PB3M       | ID[1:0]    | _          | _          | PB2M       | D[1:0]    |
|        |                  |            | _          | PB1M       | ID[1:0]    | _          |            | PB0M       | D[1:0]    |
|        | PCIOR            | PC15IOR    | PC14IOR    | PC13IOR    | PC12IOR    | PC11IOR    | PC10IOR    | PC9IOR     | PC8IOR    |
|        |                  | PC7IOR     | PC6IOR     | PC5IOR     | PC4IOR     | PC3IOR     | PC2IOR     | PC1IOR     | PC0IOR    |
|        | PCCR4            | _          | _          | _          | PC15MD     | _          | _          | _          | PC14MD    |
|        |                  |            | _          | _          | PC13MD     | _          |            | _          | PC12MD    |
|        | PCCR3            | _          | _          | _          | PC11MD     | _          | _          | _          | PC10MD    |
|        |                  |            | _          | _          | PC9MD      | _          | _          | _          | PC8MD     |
|        | PCCR2            | _          | _          | _          | PC7MD      | _          | _          |            | PC6MD     |
|        |                  |            | _          | _          | PC5MD      | _          | _          | _          | PC4MD     |
|        | PCCR1            | _          | _          | _          | PC3MD      | _          | _          | _          | PC2MD     |
|        |                  |            | _          | _          | PC1MD      | _          | _          | _          | PC0MD     |
|        | PDIOR            | _          | _          | PD13IOR    | PD12IOR    | PD11IOR    | PD10IOR    | PD9IOR     | PD8IOR    |
|        |                  | PD7IOR     | PD6IOR     | PD5IOR     | PD4IOR     | PD3IOR     | PD2IOR     | PD1IOR     | PD0IOR    |
|        | PDCR2            | _          | _          | _          | _          | PD13N      | /ID[1:0]   | PD12N      | /ID[1:0]  |
|        |                  | PD11N      | ИD[1:0]    | PD10N      | ИD[1:0]    | PD9M       | ID[1:0]    | PD8M       | D[1:0]    |
|        | PDCR1            | PD7M       | ID[1:0]    | PD6M       | ID[1:0]    | PD5M       | ID[1:0]    | PD4M       | D[1:0]    |
|        |                  | PD3M       | ID[1:0]    | PD2M       | ID[1:0]    | PD1M       | ID[1:0]    | PD0M       | D[1:0]    |
|        | PEIOR            |            | _          | PE13IOR    | PE12IOR    | PE11IOR    | PE10IOR    | PE9IOR     | PE8IOR    |
|        |                  | PE7IOR     | PE6IOR     | PE5IOR     | PE4IOR     | PE3IOR     | PE2IOR     | PE1IOR     | PE0IOR    |
|        | PECR2            |            | _          | _          | _          | _          | PE13MD     | _          | PE12MD    |
|        |                  |            | PE11MD     | _          | PE10MD     | _          | PE9MD      | _          | PE8MD     |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1  | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|------------|-----------|
| PFC    | PECR1            | PE7M       | ID[1:0]    | PE6M       | 1D[1:0]    | _          | PE5MD      | _          | PE4MD     |
|        |                  | _          | PE3MD      | PE2M       | 1D[1:0]    | PE1M       | ID[1:0]    | _          | PE0MD     |
|        | PFIOR            | PF15IOR    | PF14IOR    | PF13IOR    | PF12IOR    | PF11IOR    | PF10IOR    | PF9IOR     | PF8IOR    |
|        |                  | PF7IOR     | PF6IOR     | PF5IOR     | PF4IOR     | PF3IOR     | PF2IOR     | PF1IOR     | PF0IOR    |
|        | PFCR2            | PF15N      | /ID[1:0]   | PF14N      | ИD[1:0]    | PF13N      | MD[1:0]    | PF12M      | 1D[1:0]   |
|        |                  | PF11N      | /ID[1:0]   | PF10N      | ИD[1:0]    | PF9N       | ID[1:0]    | PF8M       | D[1:0]    |
|        | PFCR1            | PF7M       | ID[1:0]    | PF6N       | ID[1:0]    | PF5N       | D[1:0]     | PF4M       | D[1:0]    |
|        |                  | PF3M       | ID[1:0]    | PF2N       | ID[1:0]    | PF1M       | ID[1:0]    | PF0M       | D[1:0]    |
|        | PGIOR            | PG15IOR    | PG14IOR    | PG13IOR    | PG12IOR    | PG11IOR    | PG10IOR    | PG9IOR     | PG8IOR    |
|        |                  | PG7IOR     | PG6IOR     | PG5IOR     | PG4IOR     | PG3IOR     | PG2IOR     | PG1IOR     | PG0IOR    |
|        | PGCR2            | PG15N      | MD[1:0]    | PG14l      | MD[1:0]    | PG13N      | MD[1:0]    | PG12N      | MD[1:0]   |
|        |                  | PG11N      | MD[1:0]    | PG10l      | MD[1:0]    | PG9M       | ID[1:0]    | PG8M       | D[1:0]    |
|        | PGCR1            | PG7MD[1:0] |            | PG6M       | 1D[1:0]    | PG5MD[1:0] |            | PG4M       | D[1:0]    |
|        |                  | PG3M       | PG3MD[1:0] |            | PG2MD[1:0] |            | PG1MD[1:0] |            | D[1:0]    |
|        | PHIOR            | _          | _          | _          | _          | _          | _          | _          | _         |
|        |                  |            | _          | PH5IOR     | PH4IOR     | PH3IOR     | PH2IOR     | PH1IOR     | PH0IOR    |
|        | PHCR             | _          | _          |            |            | PH5MD[1:0] |            | PH4MD[1:0] |           |
|        |                  | PH3M       | ID[1:0]    | PH2M       | 1D[1:0]    | PH1MD[1:0] |            | PH0M       | D[1:0]    |
|        | PJIOR            | _          | _          | _          | _          | _          | _          | PJ9IOR     | PJ8IOR    |
|        |                  | PJ7IOR     | PJ6IOR     | PJ5IOR     | PJ4IOR     | PJ3IOR     | PJ2IOR     | PJ1IOR     | PJ0IOR    |
|        | PJCR2            | _          |            | _          |            |            |            | _          | _         |
|        |                  |            | _          | _          | _          | _          | PJ9MD      | _          | PJ8MD     |
|        | PJCR1            | PJ7M       | D[1:0]     | _          | PJ6MD      | _          | PJ5MD      | PJ4M       | D[1:0]    |
|        |                  | PJ3M       | D[1:0]     | PJ2M       | ID[1:0]    | PJ1M       | D[1:0]     | PJ0M       | D[1:0]    |
|        | PKIOR            | _          | _          | _          | _          | PK11IOR    | PK10IOR    | PK9IOR     | PK8IOR    |
|        |                  | PK7IOR     | PK6IOR     | PK5IOR     | PK4IOR     | PK3IOR     | PK2IOR     | PK1IOR     | PK0IOR    |
|        | PKCR2            | _          | _          | _          | _          | _          | _          | _          | _         |
|        |                  | PK11N      | /ID[1:0]   | PK10N      | MD[1:0]    | PK9MD[1:0] |            | _          | PK8MD     |
|        | PKCR1            | _          | PK7MD      | _          | PK6MD      | PK5MD[1:0] |            | PK4MD[1:0] |           |
|        |                  | PK3N       | ID[1:0]    | PK2N       | 1D[1:0]    | PK1N       | ID[1:0]    | PK0M       | D[1:0]    |

| Module   | Register<br>Name | 31/23/15/7 | 30/22/14/6   | 29/21/13/5 | 28/20/12/4      | 27/19/11/3 | 26/18/10/2          | 25/17/9/1      | 24/16/8/0      |
|----------|------------------|------------|--------------|------------|-----------------|------------|---------------------|----------------|----------------|
| PFC      | PLIOR            |            |              | _          |                 | _          | _                   | 20,11,0,1      | PL8IOR         |
|          | 1 21011          | PL7IOR     | PL6IOR       | PL5IOR     | PL4IOR          | PL3IOR     | PL2IOR              | PL1IOR         | PLOIOR         |
|          | PLCR2            |            | _            | _          | _               |            |                     | _              |                |
|          | . 202            |            |              |            |                 |            |                     |                | PL8MD          |
|          | PLCR1            | PI 7M      | I<br>ID[1:0] | PI 6M      | I<br>ID[1:0]    | PI 5M      | <u> </u><br> D[1:0] | PI 4M          | D[1:0]         |
|          | 1.20             |            | ID[1:0]      |            | ID[1:0]         |            | ID[1:0]             |                | D[1:0]         |
| I/O port | PADR             | PA15DR     | PA14DR       | PA13DR     | PA12DR          | PA11DR     | PA10DR              | PA9DR          | PA8DR          |
| , o port | T ALST T         | PA7DR      | PA6DR        | PA5DR      | PA4DR           | PA3DR      | PA2DR               | PA1DR          | PA0DR          |
|          | PAPR             | PA15PR     | PA14PR       | PA13PR     | PA12PR          | PA11PR     | PA10PR              | PA9PR          | PA8PR          |
|          |                  | PA7PR      | PA6PR        | PA5PR      | PA4PR           | PA3PR      | PA2PR               | PA1PR          | PA0PR          |
|          | PBDR             |            | PB14DR       | PB13DR     | PB12DR          | PB11DR     | PB10DR              | PB9DR          | PB8DR          |
|          | 1 5511           | PB7DR      | PB6DR        | PB5DR      | PB4DR           | PB3DR      | PB2DR               | PB1DR          | PB0DR          |
|          | PBPR             |            | PB14PR       | PB13PR     | PB12PR          | PB11PR     | PB10PR              | PB9PR          | PB8PR          |
|          | I DI II          | PB7PR      | PB6PR        | PB5PR      | PB4PR           | PB3PR      | PB2PR               | PB1PR          | PB0PR          |
|          | PBIR             |            |              | PB13IR     | PB12IR          | PB11IR     |                     |                | PB8IR          |
|          | I Bii i          |            |              | T B TOIL   | PB4IR           | PB3IR      | PB2IR               | PB1IR          | PB0IR          |
|          | PBDSR            |            |              | PB13DSR    | PB12DSR         |            |                     | _              |                |
|          | I BBOIT          |            |              | _          | PB4DSR          | PB3DSR     | PB2DSR              | PB1DSR         | PB0DSR         |
|          | PBPSR            |            |              |            |                 |            |                     |                | _ Bobon        |
|          | I BI GIT         |            |              |            |                 | PB3PSR     |                     | PB1PSR         |                |
|          | PCDR             | PC15DR     | PC14DR       | PC13DR     | PC12DR          | PC11DR     | PC10DR              | PC9DR          | PC8DR          |
|          | I OBIT           | PC7DR      | PC6DR        | PC5DR      | PC4DR           | PC3DR      | PC2DR               | PC1DR          | PC0DR          |
|          | PCPR             | PC15PR     | PC14PR       | PC13PR     | PC12PR          | PC11PR     | PC10PR              | PC9PR          | PC8PR          |
|          | 10111            | PC7PR      | PC6PR        | PC5PR      | PC4PR           | PC3PR      | PC2PR               | PC1PR          | PC0PR          |
|          | PDDR             | 107111     | 1 001 11     | PD13DR     | PD12DR          | PD11DR     | PD10DR              | PD9DR          | PD8DR          |
|          | I DDIT           | PD7DR      | PD6DR        | PD5DR      | PD4DR           | PD3DR      | PD2DR               | PD1DR          | PD0DR          |
|          | PDPR             |            | _ Dobn       | PD13PR     | PD12PR          | PD11PR     | PD10PR              | PD9PR          | PD8PR          |
|          |                  | PD7PR      | PD6PR        | PD5PR      | PD4PR           | PD3PR      | PD2PR               | PD1PR          | PD0PR          |
|          | PDIR             | FU/FN      | LDOLU        | LDOLU      | PD4PR           | PD11IR     | PD10IR              | PD9IR          | PD0PR<br>PD8IR |
|          | FUIK             | PD7IR      | PD6IR        | PD5IR      | PD12IR<br>PD4IR | PD3IR      | PD10IR<br>PD2IR     | PD9IR<br>PD1IR | PD8IR<br>PD0IR |
|          |                  | PD/IK      | LDOIK        | LDOIK      | PD4IK           | PUSIK      | LUZIK               | PUIK           | POUR           |

| Module   | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|----------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| I/O port | PEDR             | _          | _          | PE13DR     | PE12DR     | PE11DR     | PE10DR     | PE9DR     | PE8DR     |
|          |                  | PE7DR      | PE6DR      | PE5DR      | PE4DR      | PE3DR      | PE2DR      | PE1DR     | PE0DR     |
|          | PEPR             | _          | _          | PE13PR     | PE12PR     | PE11PR     | PE10PR     | PE9PR     | PE8PR     |
|          |                  | PE7PR      | PE6PR      | PE5PR      | PE4PR      | PE3PR      | PE2PR      | PE1PR     | PE0PR     |
|          | PEIR             | _          | _          | PE13IR     | PE12IR     | PE11IR     | PE10IR     | PE9IR     | PE8IR     |
|          |                  | PE7IR      | PE6IR      | _          | _          | _          | PE2IR      | PE1IR     | _         |
|          | PEDSR            | _          | _          | _          | _          | _          | PE10DSR    | PE9DSR    | PE8DSR    |
|          |                  | PE7DSR     | PE6DSR     | _          | _          | _          | _          | _         | _         |
|          | PFDR             | PF15DR     | PF14DR     | PF13DR     | PF12DR     | PF11DR     | PF10DR     | PF9DR     | PF8DR     |
|          |                  | PF7DR      | PF6DR      | PF5DR      | PF4DR      | PF3DR      | PF2DR      | PF1DR     | PF0DR     |
|          | PFPR             | PF15PR     | PF14PR     | PF13PR     | PF12PR     | PF11PR     | PF10PR     | PF9PR     | PF8PR     |
|          |                  | PF7PR      | PF6PR      | PF5PR      | PF4PR      | PF3PR      | PF2PR      | PF1PR     | PF0PR     |
|          | PFIR             | PF15IR     | PF14IR     | PF13IR     | PF12IR     | PF11IR     | PF10IR     | PF9IR     | PF8IR     |
|          |                  | PF7IR      | PF6IR      | PF5IR      | PF4IR      | PF3IR      | PF2IR      | PF1IR     | PF0IR     |
|          | PFDSR            | _          | PF14DSR    |            |            |            |            | _         | _         |
|          |                  | _          |            |            | _          | _          |            | _         | _         |
|          | PFPSR            | PF15PSR    | _          | _          | _          | _          | _          | _         |           |
|          |                  | _          | _          | _          | _          | _          | _          | _         | _         |
|          | PGDR             | PG15DR     | PG14DR     | PG13DR     | PG12DR     | PG11DR     | PG10DR     | PG9DR     | PG8DR     |
|          |                  | PG7DR      | PG6DR      | PG5DR      | PG4DR      | PG3DR      | PG2DR      | PG1DR     | PG0DR     |
|          | PGPR             | PG15PR     | PG14PR     | PG13PR     | PG12PR     | PG11PR     | PG10PR     | PG9PR     | PG8PR     |
|          |                  | PG7PR      | PG6PR      | PG5PR      | PG4PR      | PG3PR      | PG2PR      | PG1PR     | PG0PR     |
|          | PGIR             | PG15IR     | PG14IR     | PG13IR     | PG12IR     | PG11IR     | PG10IR     | PG9IR     | PG8IR     |
|          |                  | PG7IR      | PG6IR      | PG5IR      | PG4IR      | PG3IR      | PG2IR      | PG1IR     | PG0IR     |
|          | PGDSR            | PG15DSR    | PG14DSR    | PG13DSR    | PG12DSR    | PG11DSR    | PG10DSR    | PG9DSR    | PG8DSR    |
|          |                  | PG7DSR     | PG6DSR     | PG5DSR     | PG4DSR     | PG3DSR     | PG2DSR     | PG1DSR    | PG0DSR    |
|          | PGER             |            |            |            |            |            |            | PGHE      | S[1:0]    |
|          |                  |            |            |            |            |            |            | PGLE      | S[1:0]    |
|          | PHDR             |            |            |            |            |            |            |           |           |
|          |                  |            |            | PH5DR      | PH4DR      | PH3DR      | PH2DR      | PH1DR     | PH0DR     |

| Module   | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|----------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| I/O port | PHPR             | _          |            | _          | _          | _          |            | _         | _         |
|          |                  |            |            | PH5PR      | PH4PR      | PH3PR      | PH2PR      | PH1PR     | PH0PR     |
|          | PJDR             | _          | _          | _          | _          | _          | _          | PJ9DR     | PJ8DR     |
|          |                  | PJ7DR      | PJ6DR      | PJ5DR      | PJ4DR      | PJ3DR      | PJ2DR      | PJ1DR     | PJ0DR     |
|          | PJPR             | _          | _          | _          | _          | _          | _          | PJ9PR     | PJ8PR     |
|          |                  | PJ7PR      | PJ6PR      | PJ5PR      | PJ4PR      | PJ3PR      | PJ2PR      | PJ1PR     | PJ0PR     |
|          | PJIR             | _          | _          | _          | _          | _          | _          | _         | PJ8IR     |
|          |                  | PJ7IR      | _          | PJ5IR      | PJ4IR      | _          | PJ2IR      | _         | PJ0IR     |
|          | PJDSR            | _          | _          | _          | _          | _          |            | _         | PJ8DSR    |
|          |                  | PJ7DSR     | _          | PJ5DSR     | PJ4DSR     | _          | PJ2DSR     | _         | PJ0DSR    |
|          | PJPSR            | _          | _          | _          | _          | _          | _          | PJ9PSR    |           |
|          |                  |            | PJ6PSR     | _          | _          | PJ3PSR     |            | PJ1PSR    | _         |
|          | PKDR             | _          |            | _          | _          | PK11DR     | PK10DR     | PK9DR     | PK8DR     |
|          |                  | PK7DR      | PK6DR      | PK5DR      | PK4DR      | PK3DR      | PK2DR      | PK1DR     | PK0DR     |
|          | PKPR             | _          | _          | _          | _          | PK11PR     | PK10PR     | PK9PR     | PK8PR     |
|          |                  | PK7PR      | PK6PR      | PK5PR      | PK4PR      | PK3PR      | PK2PR      | PK1PR     | PK0PR     |
|          | PKIR             | _          | _          | _          | _          | PK11IR     | PK10IR     | PK9IR     | _         |
|          |                  | PK7IR      | PK6IR      | PK5IR      | PK4IR      | PK3IR      | PK2IR      | PK1IR     | PK0IR     |
|          | PKDSR            | _          |            | _          | _          | PK11DSR    | PK10DSR    | PK9DSR    | _         |
|          |                  | PK7DSR     | PK6DSR     | PK5DSR     | PK4DSR     | PK3DSR     | PK2DSR     | PK1DSR    | PK0DSR    |
|          | PKPSR            | _          | _          | _          | _          | PK11PSR    | _          | _         | PK8PSR    |
|          |                  |            |            | PK5PSR     | _          | _          | PK2PSR     | _         | _         |
|          | PLDR             | _          |            | _          | _          | _          |            | _         | PL8DR     |
|          |                  | PL7DR      | PL6DR      | PL5DR      | PL4DR      | PL3DR      | PL2DR      | PL1DR     | PL0DR     |
|          | PLPR             | _          |            | _          | _          | _          |            | _         | PL8PR     |
|          |                  | PL7PR      | PL6PR      | PL5PR      | PL4PR      | PL3PR      | PL2PR      | PL1PR     | PL0PR     |
|          | PLIR             |            | _          | _          | _          | _          | _          | _         | PL8IR     |
|          |                  | PL7IR      | PL6IR      | PL5IR      | PL4IR      | PL3IR      | PL2IR      | PL1IR     | _         |
|          | CKCR             |            | _          | _          | _          | _          |            | _         | _         |
|          |                  |            | _          | _          | _          | _          | _          | _         | CKOE      |

| Module | Register<br>Name | 31/23/15/7    | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |  |
|--------|------------------|---------------|--|------------|------------|------------|------------|-----------|-----------|--|
| MISG   | MISRCDR          |               |  | I          | MISRCI     | D[31:24]   | I          | I         | I         |  |
|        |                  | MISRCD[23:16] |  |            |            |            |            |           |           |  |
|        |                  |               | MISRCD[15:8]   |            |            |            |            |           |           |  |
|        |                  |               | MISRCD[7:0]  |            |            |            |            |           |           |  |
|        | MISRCR           | _             | MISREI   |            |            |            |            |           |           |  |
|        | MISR             |               |  | l          | MISR[      | 31:24]     | l          |           | 1         |  |
|        |                  |               |  |            | MISR[      | 23:16]     |            |           |           |  |
|        |                  |               |  |            | MISR       | [15:8]     |            |           |           |  |
|        |                  |               |  |            | MISF       | R[7:0]     |            |           |           |  |
| ROM/   | FPMON            | FWE           | _  | _          | _          |            | _          | _         |           |  |
| EEPROM | FMODR            | _             | _  | _          | FRDMD      | _          | _          | _         | _         |  |
|        | FASTAT           | ROMAE         | _  | _          | CMDLK      | EEPAE      | EEPIFE     | EEPRPE    | EEPWPE    |  |
|        | FAEINT           | ROMAEIE       | ROMAEIE - CMDLKIE EEPAEIE EEPIFEIE EEPRPEIE EEPWPEIE |            |            |            |            |           |           |  |
|        | ROMMAT           | KEY           |  |            |            |            |            |           | •         |  |
|        |                  | _             | _  | _          | _          |            | _          | _         | ROMSEL    |  |
|        | FCURAME          | KEY           |  |            |            |            |            |           |           |  |
|        |                  | _             |  | _          | _          |            | _          | _         | FCRME     |  |
|        | FSTATR0          | FRDY          | ILGLERR  | ERSERR     | PRGERR     | SUSRDY     | _          | ERSSPD    | PRGSPD    |  |
|        | FSTATR1          | FCUERR        |  |            | FLOCKST    | _          |            | FRDTCT    | FRCRCT    |  |
|        | FENTRYR          |               |  |            | FER        | ŒΥ         |            |           |           |  |
|        |                  | FENTRYD       | _  | FENTRY5    | FENTRY4    | FENTRY3    | FENTRY2    | FENTRY1   | FENTRY0   |  |
|        | FPROTR           |               |  |            | FPk        | (EY        |            |           |           |  |
|        |                  | _             |  |            | _          | _          |            |           | FPROTCN   |  |
|        | FRESETR          |               |  |            | FR         | KEY        |            |           |           |  |
|        |                  |               |  |            | _          |            |            |           | FRESET    |  |
|        | FCMDR            |               |  |            | СМ         | DR         |            |           |           |  |
|        |                  |               |  |            | PCN        | /IDR       |            |           |           |  |
|        | FRAMECCR         |               |  | _          | _          |            | _          | FRDCLE    | FRCCLE    |  |
|        | FCPSR            |               | _  |            |            |            |            |           |           |  |
|        |                  |               |  |            |            |            |            |           | ESUSPMD   |  |

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| ROM/   | EEPBCCNT         |            | _          | _          |            | 1          | BCADR      | •         | ·I        |
| EEPROM |                  |            |            | BCADR      |            |            | _          | _         | BCSIZE    |
|        | FPESTAT          |            | _          | _          | _          | _          | _          | _         | _         |
|        |                  | PEERRST    |            |            |            |            |            |           |           |
|        | EEPBCSTAT        | _          | _          | _          | _          | _          | _          | _         |           |
|        |                  | _          | _          | _          | _          | _          | _          | _         | BCST      |
|        | EEPRE0           | KEY        |            |            |            |            |            | •         | •         |
|        |                  | DBRE07     | DBRE06     | DBRE05     | DBRE04     | DBRE03     | DBRE02     | DBRE01    | DBRE00    |
|        | EEPRE1           |            | •          | •          | KI         | EY         | •          | •         | •         |
|        |                  | DBRE15     | DBRE14     | DBRE13     | DBRE12     | DBRE11     | DBRE10     | DBRE09    | DBRE08    |
|        | EEPWE0           |            | •          | •          | KI         | EY         | •          | •         | •         |
|        |                  | DBWE07     | DBWE06     | DBWE05     | DBWE04     | DBWE03     | DBWE02     | DBWE01    | DBWE00    |
|        | EEPWE1           | KEY        |            |            |            |            |            | •         | II.       |
|        |                  | DBWE15     | DBWE14     | DBWE13     | DBWE12     | DBWE11     | DBWE10     | DBWE09    | DBWE08    |
|        | EEPMAT           | KEY        |            |            |            |            |            | •         | •         |
|        |                  | _          | _          | _          | _          | _          | _          | _         | EEPSEL    |
| ROMC   | RCCR             | _          | _          | _          | _          | _          | _          | _         |           |
|        |                  | _          | _          | _          | _          | _          | _          | _         | _         |
|        |                  |            | _          | _          | _          | _          | _          | _         |           |
|        |                  | _          | _          | _          | _          | RCF        | RCFI       | RCFD      | RCE       |
|        | RCCR2            | _          | _          | _          | _          | _          | _          | _         | _         |
|        |                  | _          | _          | _          | _          | _          | _          | _         |           |
|        |                  | _          | _          | _          | _          | _          | _          | _         |           |
|        |                  | PFECB      | PFENB      | PFECF      | PFE        | _          | PCE2       | _         | PCE0      |
| RAM    | RAMEN0           |            |            |            | RN0KI      | EY[7:0]    |            |           |           |
|        |                  | RAME7      | RAME6      | RAME5      | RAME4      | RAME3      | RAME2      | RAME1     | RAME0     |
|        | RAMWEN0          |            | -          | -          | RWN0k      | KEY[7:0]   | -          |           |           |
|        |                  | RAMWE7     | RAMWE6     | RAMWE5     | RAMWE4     | RAMWE3     | RAMWE2     | RAMWE1    | RAMWEO    |
|        | RAMEN1           |            |            |            | RN1KI      | EY[7:0]    |            |           |           |
|        |                  | RAME15     | RAME14     | RAME13     | RAME12     | RAME11     | RAME10     | RAME9     | RAME8     |

| Module     | Register<br>Name | 31/23/15/7 | 30/22/14/6   | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|------------|------------------|------------|--------------|------------|------------|------------|------------|-----------|-----------|
| RAM        | RAMWEN1          |            | RWN1KEY[7:0] |            |            |            |            |           |           |
|            |                  | RAMWE15    | RAMWE14      | RAMWE13    | RAMWE12    | RAMWE11    | RAMWE10    | RAMWE9    | RAMWE8    |
|            | RAMECC           |            | REKEY[7:0]   |            |            |            |            |           |           |
|            |                  | _          |              |            |            | _          |            |           | RECCA     |
|            | RAMERR           |            | _            | _          | RPARI      | _          | _          | RDTCT     | RCRCT     |
|            | RAMINT           |            |              | RPEIE1     | RPEIE0     | REDIE1     | RECIE1     | REDIE0    | RECIE0    |
|            | RAMACYC          |            | RAKEY[7:0]   |            |            |            |            |           |           |
|            |                  |            |              | WRC        | /C[1:0]    |            | _          | _         | RDCYC     |
| Power-down | STBCR            | STBCRKEY   |              |            |            |            |            |           |           |
| mode       |                  | _          |              |            | MSTP4      | MSTP3      | MSTP2      | MSTP1     | MSTP0     |

## 31.3 Register States in Each Operating Mode

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| INTC   | ICR0          | Initialized | Initialized         | Retained     |
|        | ICR1          | Initialized | Initialized         | Retained     |
|        | IRQRR         | Initialized | Initialized         | Retained     |
|        | IBCR          | Initialized | Initialized         | Retained     |
|        | IBNR          | Initialized | Initialized         | Retained     |
|        | SINTR1        | Initialized | Initialized         | Retained     |
|        | SINTR2        | Initialized | Initialized         | Retained     |
|        | SINTR3        | Initialized | Initialized         | Retained     |
|        | SINTR4        | Initialized | Initialized         | Retained     |
|        | SINTR5        | Initialized | Initialized         | Retained     |
|        | SINTR6        | Initialized | Initialized         | Retained     |
|        | SINTR7        | Initialized | Initialized         | Retained     |
|        | SINTR8        | Initialized | Initialized         | Retained     |
|        | IPR01         | Initialized | Initialized         | Retained     |
|        | IPR02         | Initialized | Initialized         | Retained     |
|        | SINTR9        | Initialized | Initialized         | Retained     |
|        | SINTR10       | Initialized | Initialized         | Retained     |
|        | SINTR11       | Initialized | Initialized         | Retained     |
|        | SINTR12       | Initialized | Initialized         | Retained     |
|        | SINTR13       | Initialized | Initialized         | Retained     |
|        | SINTR14       | Initialized | Initialized         | Retained     |
|        | SINTR15       | Initialized | Initialized         | Retained     |
|        | IPR03         | Initialized | Initialized         | Retained     |
|        | IPR04         | Initialized | Initialized         | Retained     |
|        | IPR05         | Initialized | Initialized         | Retained     |
|        | IPR06         | Initialized | Initialized         | Retained     |
|        | IPR07         | Initialized | Initialized         | Retained     |

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| INTC   | IPR08         | Initialized | Initialized         | Retained     |
|        | IPR09         | Initialized | Initialized         | Retained     |
|        | IPR10         | Initialized | Initialized         | Retained     |
|        | IPR11         | Initialized | Initialized         | Retained     |
|        | IPR12         | Initialized | Initialized         | Retained     |
|        | IPR13         | Initialized | Initialized         | Retained     |
|        | IPR14         | Initialized | Initialized         | Retained     |
|        | IPR15         | Initialized | Initialized         | Retained     |
|        | IPR16         | Initialized | Initialized         | Retained     |
|        | IPR17         | Initialized | Initialized         | Retained     |
|        | IPR18         | Initialized | Initialized         | Retained     |
|        | IPR19         | Initialized | Initialized         | Retained     |
|        | IPR20         | Initialized | Initialized         | Retained     |
|        | IPR21         | Initialized | Initialized         | Retained     |
|        | IPR22         | Initialized | Initialized         | Retained     |
|        | IPR23         | Initialized | Initialized         | Retained     |
|        | IPR24         | Initialized | Initialized         | Retained     |
|        | IPR25         | Initialized | Initialized         | Retained     |
|        | IPR26         | Initialized | Initialized         | Retained     |
|        | IPR27         | Initialized | Initialized         | Retained     |
|        | IPR28         | Initialized | Initialized         | Retained     |
|        | IPR29         | Initialized | Initialized         | Retained     |
| UBC    | BAR_0         | Initialized | Initialized         | Retained     |
|        | BAMR_0        | Initialized | Initialized         | Retained     |
|        | BBR_0         | Initialized | Initialized         | Retained     |
|        | BAR_1         | Initialized | Initialized         | Retained     |
|        | BAMR_1        | Initialized | Initialized         | Retained     |
|        | BBR_1         | Initialized | Initialized         | Retained     |
|        | BAR_2         | Initialized | Initialized         | Retained     |

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| UBC    | BAMR_2        | Initialized | Initialized         | Retained     |
|        | BBR_2         | Initialized | Initialized         | Retained     |
|        | BAR_3         | Initialized | Initialized         | Retained     |
|        | BAMR_3        | Initialized | Initialized         | Retained     |
|        | BBR_3         | Initialized | Initialized         | Retained     |
|        | BRCR          | Initialized | Initialized         | Retained     |
| BSC    | CS0BCR        | Initialized | Initialized         | Retained     |
|        | CS1BCR        | Initialized | Initialized         | Retained     |
|        | CS2BCR        | Initialized | Initialized         | Retained     |
|        | CS3BCR        | Initialized | Initialized         | Retained     |
|        | CS0WCR        | Initialized | Initialized         | Retained     |
|        | CS1WCR        | Initialized | Initialized         | Retained     |
|        | CS2WCR        | Initialized | Initialized         | Retained     |
|        | CS3WCR        | Initialized | Initialized         | Retained     |
| DMAC   | SAR0          | Initialized | Initialized         | Retained     |
|        | DAR0          | Initialized | Initialized         | Retained     |
|        | DMATCR0       | Initialized | Initialized         | Retained     |
|        | CHCR0         | Initialized | Initialized         | Retained     |
|        | CHFR0         | Initialized | Initialized         | Retained     |
|        | TEMSK0        | Initialized | Initialized         | Retained     |
|        | RSAR0         | Initialized | Initialized         | Retained     |
|        | RDAR0         | Initialized | Initialized         | Retained     |
|        | RDMATCR0      | Initialized | Initialized         | Retained     |
|        | SAR1          | Initialized | Initialized         | Retained     |
|        | DAR1          | Initialized | Initialized         | Retained     |
|        | DMATCR1       | Initialized | Initialized         | Retained     |
|        | CHCR1         | Initialized | Initialized         | Retained     |
|        | CHFR1         | Initialized | Initialized         | Retained     |
|        | TEMSK1        | Initialized | Initialized         | Retained     |
|        | RSAR1         | Initialized | Initialized         | Retained     |

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| DMAC   | RDAR1         | Initialized | Initialized         | Retained     |
|        | RDMATCR1      | Initialized | Initialized         | Retained     |
|        | SAR2          | Initialized | Initialized         | Retained     |
|        | DAR2          | Initialized | Initialized         | Retained     |
|        | DMATCR2       | Initialized | Initialized         | Retained     |
|        | CHCR2         | Initialized | Initialized         | Retained     |
|        | CHFR2         | Initialized | Initialized         | Retained     |
|        | TEMSK2        | Initialized | Initialized         | Retained     |
|        | RSAR2         | Initialized | Initialized         | Retained     |
|        | RDAR2         | Initialized | Initialized         | Retained     |
|        | RDMATCR2      | Initialized | Initialized         | Retained     |
|        | SAR3          | Initialized | Initialized         | Retained     |
|        | DAR3          | Initialized | Initialized         | Retained     |
|        | DMATCR3       | Initialized | Initialized         | Retained     |
|        | CHCR3         | Initialized | Initialized         | Retained     |
|        | CHFR3         | Initialized | Initialized         | Retained     |
|        | TEMSK3        | Initialized | Initialized         | Retained     |
|        | RSAR3         | Initialized | Initialized         | Retained     |
|        | RDAR3         | Initialized | Initialized         | Retained     |
|        | RDMATCR3      | Initialized | Initialized         | Retained     |
|        | SAR4          | Initialized | Initialized         | Retained     |
|        | DAR4          | Initialized | Initialized         | Retained     |
|        | DMATCR4       | Initialized | Initialized         | Retained     |
|        | CHCR4         | Initialized | Initialized         | Retained     |
|        | CHFR4         | Initialized | Initialized         | Retained     |
|        | TEMSK4        | Initialized | Initialized         | Retained     |
|        | RSAR4         | Initialized | Initialized         | Retained     |
|        | RDAR4         | Initialized | Initialized         | Retained     |
|        | RDMATCR4      | Initialized | Initialized         | Retained     |
|        | ARCR4         | Initialized | Initialized         | Retained     |

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| DMAC   | RARCR4        | Initialized | Initialized         | Retained     |
|        | SAR5          | Initialized | Initialized         | Retained     |
|        | DAR5          | Initialized | Initialized         | Retained     |
|        | DMATCR5       | Initialized | Initialized         | Retained     |
|        | CHCR5         | Initialized | Initialized         | Retained     |
|        | CHFR5         | Initialized | Initialized         | Retained     |
|        | TEMSK5        | Initialized | Initialized         | Retained     |
|        | RSAR5         | Initialized | Initialized         | Retained     |
|        | RDAR5         | Initialized | Initialized         | Retained     |
|        | RDMATCR5      | Initialized | Initialized         | Retained     |
|        | ARCR5         | Initialized | Initialized         | Retained     |
|        | RARCR5        | Initialized | Initialized         | Retained     |
|        | SAR6          | Initialized | Initialized         | Retained     |
|        | DAR6          | Initialized | Initialized         | Retained     |
|        | DMATCR6       | Initialized | Initialized         | Retained     |
|        | CHCR6         | Initialized | Initialized         | Retained     |
|        | CHFR6         | Initialized | Initialized         | Retained     |
|        | TEMSK6        | Initialized | Initialized         | Retained     |
|        | RSAR6         | Initialized | Initialized         | Retained     |
|        | RDAR6         | Initialized | Initialized         | Retained     |
|        | RDMATCR6      | Initialized | Initialized         | Retained     |
|        | ARCR6         | Initialized | Initialized         | Retained     |
|        | RARCR6        | Initialized | Initialized         | Retained     |
|        | SAR7          | Initialized | Initialized         | Retained     |
|        | DAR7          | Initialized | Initialized         | Retained     |
|        | DMATCR7       | Initialized | Initialized         | Retained     |
|        | CHCR7         | Initialized | Initialized         | Retained     |
|        | CHFR7         | Initialized | Initialized         | Retained     |
|        | TEMSK7        | Initialized | Initialized         | Retained     |
|        | RSAR7         | Initialized | Initialized         | Retained     |

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| DMAC   | RDAR7         | Initialized | Initialized         | Retained     |
|        | RDMATCR7      | Initialized | Initialized         | Retained     |
|        | ARCR7         | Initialized | Initialized         | Retained     |
|        | RARCR7        | Initialized | Initialized         | Retained     |
|        | DMAOR         | Initialized | Initialized         | Retained     |
|        | DMAFR         | Initialized | Initialized         | Retained     |
|        | DMARS0        | Initialized | Initialized         | Retained     |
|        | DMARS1        | Initialized | Initialized         | Retained     |
|        | DMARS2        | Initialized | Initialized         | Retained     |
|        | DMARS3        | Initialized | Initialized         | Retained     |
| A-DMAC | ADMAOR        | Initialized | Initialized         | Retained     |
|        | ADMAABR       | Initialized | Initialized         | Retained     |
|        | ADMAIE0       | Initialized | Initialized         | Retained     |
|        | ADMAIE1       | Initialized | Initialized         | Retained     |
|        | ADMAIE2       | Initialized | Initialized         | Retained     |
|        | ADMAIE3       | Initialized | Initialized         | Retained     |
|        | ADMAIE4       | Initialized | Initialized         | Retained     |
|        | ADMAIE5       | Initialized | Initialized         | Retained     |
|        | ADMAIE7       | Initialized | Initialized         | Retained     |
|        | ADMAIE8       | Initialized | Initialized         | Retained     |
|        | ADMAIE9       | Initialized | Initialized         | Retained     |
|        | ADMADV0       | Initialized | Initialized         | Retained     |
|        | ADMADV1       | Initialized | Initialized         | Retained     |
|        | ADMADV2       | Initialized | Initialized         | Retained     |
|        | ADMADV3       | Initialized | Initialized         | Retained     |
|        | ADMADV4       | Initialized | Initialized         | Retained     |
|        | ADMADV5       | Initialized | Initialized         | Retained     |
|        | ADMATE0       | Initialized | Initialized         | Retained     |
|        | ADMATE1       | Initialized | Initialized         | Retained     |
|        | ADMATE2       | Initialized | Initialized         | Retained     |

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| A-DMAC | ADMADE0       | Initialized | Initialized         | Retained     |
|        | ADMADE1       | Initialized | Initialized         | Retained     |
|        | ADMADE2       | Initialized | Initialized         | Retained     |
|        | ADMADE3       | Initialized | Initialized         | Retained     |
|        | ADMADE4       | Initialized | Initialized         | Retained     |
|        | ADMADE5       | Initialized | Initialized         | Retained     |
|        | ADMADE7       | Initialized | Initialized         | Retained     |
|        | ADMAMODE0     | Initialized | Initialized         | Retained     |
|        | ADMAMODE1     | Initialized | Initialized         | Retained     |
|        | ADMAMODE2     | Initialized | Initialized         | Retained     |
|        | ADMATCR0      | Initialized | Initialized         | Retained     |
|        | ADMARTCR0     | Initialized | Initialized         | Retained     |
|        | ADMATCR1      | Initialized | Initialized         | Retained     |
|        | ADMARTCR1     | Initialized | Initialized         | Retained     |
|        | ADMATCR56     | Initialized | Initialized         | Retained     |
|        | ADMATCR57     | Initialized | Initialized         | Retained     |
|        | ADMATCR58     | Initialized | Initialized         | Retained     |
|        | ADMATCR59     | Initialized | Initialized         | Retained     |
|        | ADMATCR60     | Initialized | Initialized         | Retained     |
|        | ADMATCR61     | Initialized | Initialized         | Retained     |
|        | ADMATCR62     | Initialized | Initialized         | Retained     |
|        | ADMATCR63     | Initialized | Initialized         | Retained     |
|        | ADMATCR64     | Initialized | Initialized         | Retained     |
|        | ADMATCR65     | Initialized | Initialized         | Retained     |
|        | ADMATCR66     | Initialized | Initialized         | Retained     |
|        | ADMATCR67     | Initialized | Initialized         | Retained     |
|        | ADMATCR68     | Initialized | Initialized         | Retained     |
|        | ADMATCR69     | Initialized | Initialized         | Retained     |
|        | ADMATCR70     | Initialized | Initialized         | Retained     |
|        | ADMATCR71     | Initialized | Initialized         | Retained     |

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| A-DMAC | ADMAAR0       | Initialized | Initialized         | Retained     |
|        | ADMARAR0      | Initialized | Initialized         | Retained     |
|        | ADMAAR1       | Initialized | Initialized         | Retained     |
|        | ADMARAR1      | Initialized | Initialized         | Retained     |
|        | ADMAAR56      | Initialized | Initialized         | Retained     |
|        | ADMAAR57      | Initialized | Initialized         | Retained     |
|        | ADMAAR58      | Initialized | Initialized         | Retained     |
|        | ADMAAR59      | Initialized | Initialized         | Retained     |
|        | ADMAAR60      | Initialized | Initialized         | Retained     |
|        | ADMAAR61      | Initialized | Initialized         | Retained     |
|        | ADMAAR62      | Initialized | Initialized         | Retained     |
|        | ADMAAR63      | Initialized | Initialized         | Retained     |
|        | ADMAAR64      | Initialized | Initialized         | Retained     |
|        | ADMAAR65      | Initialized | Initialized         | Retained     |
|        | ADMAAR66      | Initialized | Initialized         | Retained     |
|        | ADMAAR67      | Initialized | Initialized         | Retained     |
|        | ADMAAR68      | Initialized | Initialized         | Retained     |
|        | ADMAAR69      | Initialized | Initialized         | Retained     |
|        | ADMAAR70      | Initialized | Initialized         | Retained     |
|        | ADMAAR71      | Initialized | Initialized         | Retained     |
|        | ADMABUF2      | Initialized | Initialized         | Retained     |
|        | ADMABUF3      | Initialized | Initialized         | Retained     |
|        | ADMABUF4      | Initialized | Initialized         | Retained     |
|        | ADMABUF5      | Initialized | Initialized         | Retained     |
|        | ADMABUF6      | Initialized | Initialized         | Retained     |
|        | ADMABUF7      | Initialized | Initialized         | Retained     |
|        | ADMARVPR0     | Initialized | Initialized         | Retained     |
|        | ADMARVPR1     | Initialized | Initialized         | Retained     |
|        | ADMARVPR2     | Initialized | Initialized         | Retained     |
|        | ADMARVPR3     | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| A-DMAC  | ADMARVPR4     | Initialized | Initialized         | Retained     |
|         | ADMARVPR5     | Initialized | Initialized         | Retained     |
|         | ADMATVPR0     | Initialized | Initialized         | Retained     |
|         | ADMATVPR1     | Initialized | Initialized         | Retained     |
|         | ADMATVPR2     | Initialized | Initialized         | Retained     |
|         | ADMATVPR3     | Initialized | Initialized         | Retained     |
|         | ADMATVPR4     | Initialized | Initialized         | Retained     |
|         | ADMATVPR5     | Initialized | Initialized         | Retained     |
| ATU-III | ATUENR        | Initialized | Initialized         | Retained     |
|         | CBCNT         | Initialized | Initialized         | Retained     |
|         | NCMR          | Initialized | Initialized         | Retained     |
|         | PSCR0         | Initialized | Initialized         | Retained     |
|         | PSCR1         | Initialized | Initialized         | Retained     |
|         | PSCR2         | Initialized | Initialized         | Retained     |
|         | PSCR3         | Initialized | Initialized         | Retained     |
|         | TCRA          | Initialized | Initialized         | Retained     |
|         | TIOR1A        | Initialized | Initialized         | Retained     |
|         | TIOR2A        | Initialized | Initialized         | Retained     |
|         | TSRA          | Initialized | Initialized         | Retained     |
|         | TIERA         | Initialized | Initialized         | Retained     |
|         | NCNTA0        | Initialized | Initialized         | Retained     |
|         | NCRA0         | Initialized | Initialized         | Retained     |
|         | NCNTA1        | Initialized | Initialized         | Retained     |
|         | NCRA1         | Initialized | Initialized         | Retained     |
|         | NCNTA2        | Initialized | Initialized         | Retained     |
|         | NCRA2         | Initialized | Initialized         | Retained     |
|         | NCNTA3        | Initialized | Initialized         | Retained     |
|         | NCRA3         | Initialized | Initialized         | Retained     |
|         | NCNTA4        | Initialized | Initialized         | Retained     |
|         | NCRA4         | Initialized | Initialized         | Retained     |

|         |               | Reset State P |                     | Power-Down State |  |
|---------|---------------|---------------|---------------------|------------------|--|
| Module  | Register Name | Power-On      | Hardware<br>Standby | Sleep            |  |
| ATU-III | NCNTA5        | Initialized   | Initialized         | Retained         |  |
|         | NCRA5         | Initialized   | Initialized         | Retained         |  |
|         | TCNTA         | Initialized   | Initialized         | Retained         |  |
|         | ICRA0         | Initialized   | Initialized         | Retained         |  |
|         | ICRA1         | Initialized   | Initialized         | Retained         |  |
|         | ICRA2         | Initialized   | Initialized         | Retained         |  |
|         | ICRA3         | Initialized   | Initialized         | Retained         |  |
|         | ICRA4         | Initialized   | Initialized         | Retained         |  |
|         | ICRA5         | Initialized   | Initialized         | Retained         |  |
|         | TCRB          | Initialized   | Initialized         | Retained         |  |
|         | TIORB         | Initialized   | Initialized         | Retained         |  |
|         | TSRB          | Initialized   | Initialized         | Retained         |  |
|         | TIERB         | Initialized   | Initialized         | Retained         |  |
|         | TCNTB0        | Initialized   | Initialized         | Retained         |  |
|         | ICRB0         | Initialized   | Initialized         | Retained         |  |
|         | OCRB0         | Initialized   | Initialized         | Retained         |  |
|         | TCNTB1        | Initialized   | Initialized         | Retained         |  |
|         | OCRB1         | Initialized   | Initialized         | Retained         |  |
|         | ICRB1         | Initialized   | Initialized         | Retained         |  |
|         | ICRB2         | Initialized   | Initialized         | Retained         |  |
|         | LDB           | Initialized   | Initialized         | Retained         |  |
|         | RLDB          | Initialized   | Initialized         | Retained         |  |
|         | PIMR          | Initialized   | Initialized         | Retained         |  |
|         | TCNTB2        | Initialized   | Initialized         | Retained         |  |
|         | TCNTB6        | Initialized   | Initialized         | Retained         |  |
|         | OCRB6         | Initialized   | Initialized         | Retained         |  |
|         | OCRB7         | Initialized   | Initialized         | Retained         |  |
|         | TCNTB3        | Initialized   | Initialized         | Retained         |  |
|         | TCNTB4        | Initialized   | Initialized         | Retained         |  |
|         | TCNTB5        | Initialized   | Initialized         | Retained         |  |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | TCCLRB        | Initialized | Initialized         | Retained     |
|         | TSTRC         | Initialized | Initialized         | Retained     |
|         | NCCRC0        | Initialized | Initialized         | Retained     |
|         | NCCRC1        | Initialized | Initialized         | Retained     |
|         | NCCRC2        | Initialized | Initialized         | Retained     |
|         | NCCRC3        | Initialized | Initialized         | Retained     |
|         | NCCRC4        | Initialized | Initialized         | Retained     |
|         | NCNTC00       | Initialized | Initialized         | Retained     |
|         | NCNTC01       | Initialized | Initialized         | Retained     |
|         | NCNTC02       | Initialized | Initialized         | Retained     |
|         | NCNTC03       | Initialized | Initialized         | Retained     |
|         | NCRC00        | Initialized | Initialized         | Retained     |
|         | NCRC01        | Initialized | Initialized         | Retained     |
|         | NCRC02        | Initialized | Initialized         | Retained     |
|         | NCRC03        | Initialized | Initialized         | Retained     |
|         | NCNTC10       | Initialized | Initialized         | Retained     |
|         | NCNTC11       | Initialized | Initialized         | Retained     |
|         | NCNTC12       | Initialized | Initialized         | Retained     |
|         | NCNTC13       | Initialized | Initialized         | Retained     |
|         | NCRC10        | Initialized | Initialized         | Retained     |
|         | NCRC11        | Initialized | Initialized         | Retained     |
|         | NCRC12        | Initialized | Initialized         | Retained     |
|         | NCRC13        | Initialized | Initialized         | Retained     |
|         | NCNTC20       | Initialized | Initialized         | Retained     |
|         | NCNTC21       | Initialized | Initialized         | Retained     |
|         | NCNTC22       | Initialized | Initialized         | Retained     |
|         | NCNTC23       | Initialized | Initialized         | Retained     |
|         | NCRC20        | Initialized | Initialized         | Retained     |
|         | NCRC21        | Initialized | Initialized         | Retained     |
|         | NCRC22        | Initialized | Initialized         | Retained     |

|         |               | Reset State Power |                     | er-Down State |  |
|---------|---------------|-------------------|---------------------|---------------|--|
| Module  | Register Name | Power-On          | Hardware<br>Standby | Sleep         |  |
| ATU-III | NCRC23        | Initialized       | Initialized         | Retained      |  |
|         | NCNTC30       | Initialized       | Initialized         | Retained      |  |
|         | NCNTC31       | Initialized       | Initialized         | Retained      |  |
|         | NCNTC32       | Initialized       | Initialized         | Retained      |  |
|         | NCNTC33       | Initialized       | Initialized         | Retained      |  |
|         | NCRC30        | Initialized       | Initialized         | Retained      |  |
|         | NCRC31        | Initialized       | Initialized         | Retained      |  |
|         | NCRC32        | Initialized       | Initialized         | Retained      |  |
|         | NCRC33        | Initialized       | Initialized         | Retained      |  |
|         | NCNTC40       | Initialized       | Initialized         | Retained      |  |
|         | NCNTC41       | Initialized       | Initialized         | Retained      |  |
|         | NCNTC42       | Initialized       | Initialized         | Retained      |  |
|         | NCNTC43       | Initialized       | Initialized         | Retained      |  |
|         | NCRC40        | Initialized       | Initialized         | Retained      |  |
|         | NCRC41        | Initialized       | Initialized         | Retained      |  |
|         | NCRC42        | Initialized       | Initialized         | Retained      |  |
|         | NCRC43        | Initialized       | Initialized         | Retained      |  |
|         | TCRC0         | Initialized       | Initialized         | Retained      |  |
|         | TIERC0        | Initialized       | Initialized         | Retained      |  |
|         | TIORC0        | Initialized       | Initialized         | Retained      |  |
|         | TSRC0         | Initialized       | Initialized         | Retained      |  |
|         | GRC00         | Initialized       | Initialized         | Retained      |  |
|         | GRC01         | Initialized       | Initialized         | Retained      |  |
|         | GRC02         | Initialized       | Initialized         | Retained      |  |
|         | GRC03         | Initialized       | Initialized         | Retained      |  |
|         | TCNTC0        | Initialized       | Initialized         | Retained      |  |
|         | TCRC1         | Initialized       | Initialized         | Retained      |  |
|         | TIERC1        | Initialized       | Initialized         | Retained      |  |
|         | TIORC1        | Initialized       | Initialized         | Retained      |  |
|         | TSRC1         | Initialized       | Initialized         | Retained      |  |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | GRC10         | Initialized | Initialized         | Retained     |
|         | GRC11         | Initialized | Initialized         | Retained     |
|         | GRC12         | Initialized | Initialized         | Retained     |
|         | GRC13         | Initialized | Initialized         | Retained     |
|         | TCNTC1        | Initialized | Initialized         | Retained     |
|         | TCRC2         | Initialized | Initialized         | Retained     |
|         | TIERC2        | Initialized | Initialized         | Retained     |
|         | TIORC2        | Initialized | Initialized         | Retained     |
|         | TSRC2         | Initialized | Initialized         | Retained     |
|         | GRC20         | Initialized | Initialized         | Retained     |
|         | GRC21         | Initialized | Initialized         | Retained     |
|         | GRC22         | Initialized | Initialized         | Retained     |
|         | GRC23         | Initialized | Initialized         | Retained     |
|         | TCNTC2        | Initialized | Initialized         | Retained     |
|         | TCRC3         | Initialized | Initialized         | Retained     |
|         | TIERC3        | Initialized | Initialized         | Retained     |
|         | TIORC3        | Initialized | Initialized         | Retained     |
|         | TSRC3         | Initialized | Initialized         | Retained     |
|         | GRC30         | Initialized | Initialized         | Retained     |
|         | GRC31         | Initialized | Initialized         | Retained     |
|         | GRC32         | Initialized | Initialized         | Retained     |
|         | GRC33         | Initialized | Initialized         | Retained     |
|         | TCNTC3        | Initialized | Initialized         | Retained     |
|         | TCRC4         | Initialized | Initialized         | Retained     |
|         | TIERC4        | Initialized | Initialized         | Retained     |
|         | TIORC4        | Initialized | Initialized         | Retained     |
|         | TSRC4         | Initialized | Initialized         | Retained     |
|         | GRC40         | Initialized | Initialized         | Retained     |
|         | GRC41         | Initialized | Initialized         | Retained     |
|         | GRC42         | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | GRC43         | Initialized | Initialized         | Retained     |
|         | TCNTC4        | Initialized | Initialized         | Retained     |
|         | TSTRD         | Initialized | Initialized         | Retained     |
|         | TCNT1D0       | Initialized | Initialized         | Retained     |
|         | TCNT2D0       | Initialized | Initialized         | Retained     |
|         | OSBRD0        | Initialized | Initialized         | Retained     |
|         | TCRD0         | Initialized | Initialized         | Retained     |
|         | TOCRD0        | Initialized | Initialized         | Retained     |
|         | CMPOD0        | Initialized | Initialized         | Retained     |
|         | TCNT1D1       | Initialized | Initialized         | Retained     |
|         | TCNT2D1       | Initialized | Initialized         | Retained     |
|         | OSBRD1        | Initialized | Initialized         | Retained     |
|         | TCRD1         | Initialized | Initialized         | Retained     |
|         | TOCRD1        | Initialized | Initialized         | Retained     |
|         | CMPOD1        | Initialized | Initialized         | Retained     |
|         | TCNT1D2       | Initialized | Initialized         | Retained     |
|         | TCNT2D2       | Initialized | Initialized         | Retained     |
|         | OSBRD2        | Initialized | Initialized         | Retained     |
|         | TCRD2         | Initialized | Initialized         | Retained     |
|         | TOCRD2        | Initialized | Initialized         | Retained     |
|         | TCNT1D3       | Initialized | Initialized         | Retained     |
|         | TCNT2D3       | Initialized | Initialized         | Retained     |
|         | OSBRD3        | Initialized | Initialized         | Retained     |
|         | TCRD3         | Initialized | Initialized         | Retained     |
|         | TOCRD3        | Initialized | Initialized         | Retained     |
|         | TIOR1D0       | Initialized | Initialized         | Retained     |
|         | TIOR2D0       | Initialized | Initialized         | Retained     |
|         | DSTRD0        | Initialized | Initialized         | Retained     |
|         | DSRD0         | Initialized | Initialized         | Retained     |
|         | DCRD0         | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | TSRD0         | Initialized | Initialized         | Retained     |
|         | TIERD0        | Initialized | Initialized         | Retained     |
|         | OCRD00        | Initialized | Initialized         | Retained     |
|         | OCRD01        | Initialized | Initialized         | Retained     |
|         | OCRD02        | Initialized | Initialized         | Retained     |
|         | OCRD03        | Initialized | Initialized         | Retained     |
|         | GRD00         | Initialized | Initialized         | Retained     |
|         | GRD01         | Initialized | Initialized         | Retained     |
|         | GRD02         | Initialized | Initialized         | Retained     |
|         | GRD03         | Initialized | Initialized         | Retained     |
|         | DCNTD00       | Initialized | Initialized         | Retained     |
|         | DCNTD01       | Initialized | Initialized         | Retained     |
|         | DCNTD02       | Initialized | Initialized         | Retained     |
|         | DCNTD03       | Initialized | Initialized         | Retained     |
|         | TIOR1D1       | Initialized | Initialized         | Retained     |
|         | TIOR2D1       | Initialized | Initialized         | Retained     |
|         | DSTRD1        | Initialized | Initialized         | Retained     |
|         | DSRD1         | Initialized | Initialized         | Retained     |
|         | DCRD1         | Initialized | Initialized         | Retained     |
|         | TSRD1         | Initialized | Initialized         | Retained     |
|         | TIERD1        | Initialized | Initialized         | Retained     |
|         | OCRD10        | Initialized | Initialized         | Retained     |
|         | OCRD11        | Initialized | Initialized         | Retained     |
|         | OCRD12        | Initialized | Initialized         | Retained     |
|         | OCRD13        | Initialized | Initialized         | Retained     |
|         | GRD10         | Initialized | Initialized         | Retained     |
|         | GRD11         | Initialized | Initialized         | Retained     |
|         | GRD12         | Initialized | Initialized         | Retained     |
|         | GRD13         | Initialized | Initialized         | Retained     |
|         | DCNTD10       | Initialized | Initialized         | Retained     |

|         |               | Reset State Power-Dov |                     | r-Down State |
|---------|---------------|-----------------------|---------------------|--------------|
| Module  | Register Name | Power-On              | Hardware<br>Standby | Sleep        |
| ATU-III | DCNTD11       | Initialized           | Initialized         | Retained     |
|         | DCNTD12       | Initialized           | Initialized         | Retained     |
|         | DCNTD13       | Initialized           | Initialized         | Retained     |
|         | TIOR1D2       | Initialized           | Initialized         | Retained     |
|         | TIOR2D2       | Initialized           | Initialized         | Retained     |
|         | DSTRD2        | Initialized           | Initialized         | Retained     |
|         | DSRD2         | Initialized           | Initialized         | Retained     |
|         | DCRD2         | Initialized           | Initialized         | Retained     |
|         | TSRD2         | Initialized           | Initialized         | Retained     |
|         | TIERD2        | Initialized           | Initialized         | Retained     |
|         | OCRD20        | Initialized           | Initialized         | Retained     |
|         | OCRD21        | Initialized           | Initialized         | Retained     |
|         | OCRD22        | Initialized           | Initialized         | Retained     |
|         | OCRD23        | Initialized           | Initialized         | Retained     |
|         | GRD20         | Initialized           | Initialized         | Retained     |
|         | GRD21         | Initialized           | Initialized         | Retained     |
|         | GRD22         | Initialized           | Initialized         | Retained     |
|         | GRD23         | Initialized           | Initialized         | Retained     |
|         | DCNTD20       | Initialized           | Initialized         | Retained     |
|         | DCNTD21       | Initialized           | Initialized         | Retained     |
|         | DCNTD22       | Initialized           | Initialized         | Retained     |
|         | DCNTD23       | Initialized           | Initialized         | Retained     |
|         | TIOR1D3       | Initialized           | Initialized         | Retained     |
|         | TIOR2D3       | Initialized           | Initialized         | Retained     |
|         | DSTRD3        | Initialized           | Initialized         | Retained     |
|         | DSRD3         | Initialized           | Initialized         | Retained     |
|         | DCRD3         | Initialized           | Initialized         | Retained     |
|         | TSRD3         | Initialized           | Initialized         | Retained     |
|         | TIERD3        | Initialized           | Initialized         | Retained     |
|         | OCRD30        | Initialized           | Initialized         | Retained     |
|         | OCRD31        | Initialized           | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | OCRD32        | Initialized | Initialized         | Retained     |
|         | OCRD33        | Initialized | Initialized         | Retained     |
|         | GRD30         | Initialized | Initialized         | Retained     |
|         | GRD31         | Initialized | Initialized         | Retained     |
|         | GRD32         | Initialized | Initialized         | Retained     |
|         | GRD33         | Initialized | Initialized         | Retained     |
|         | DCNTD30       | Initialized | Initialized         | Retained     |
|         | DCNTD31       | Initialized | Initialized         | Retained     |
|         | DCNTD32       | Initialized | Initialized         | Retained     |
|         | DCNTD33       | Initialized | Initialized         | Retained     |
|         | TSTRE         | Initialized | Initialized         | Retained     |
|         | TCRE0         | Initialized | Initialized         | Retained     |
|         | TOCRE0        | Initialized | Initialized         | Retained     |
|         | TIERE0        | Initialized | Initialized         | Retained     |
|         | RLDCRE0       | Initialized | Initialized         | Retained     |
|         | TSRE0         | Initialized | Initialized         | Retained     |
|         | PSCRE0        | Initialized | Initialized         | Retained     |
|         | SSTRE0        | Initialized | Initialized         | Retained     |
|         | CYLRE00       | Initialized | Initialized         | Retained     |
|         | CYLRE01       | Initialized | Initialized         | Retained     |
|         | CYLRE02       | Initialized | Initialized         | Retained     |
|         | CYLRE03       | Initialized | Initialized         | Retained     |
|         | DTRE00        | Initialized | Initialized         | Retained     |
|         | DTRE01        | Initialized | Initialized         | Retained     |
|         | DTRE02        | Initialized | Initialized         | Retained     |
|         | DTRE03        | Initialized | Initialized         | Retained     |
|         | CRLDE00       | Initialized | Initialized         | Retained     |
|         | CRLDE01       | Initialized | Initialized         | Retained     |
|         | CRLDE02       | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | CRLDE03       | Initialized | Initialized         | Retained     |
|         | DRLDE00       | Initialized | Initialized         | Retained     |
|         | DRLDE01       | Initialized | Initialized         | Retained     |
|         | DRLDE02       | Initialized | Initialized         | Retained     |
|         | DRLDE03       | Initialized | Initialized         | Retained     |
|         | TCNTE00       | Initialized | Initialized         | Retained     |
|         | TCNTE01       | Initialized | Initialized         | Retained     |
|         | TCNTE02       | Initialized | Initialized         | Retained     |
|         | TCNTE03       | Initialized | Initialized         | Retained     |
|         | TCRE1         | Initialized | Initialized         | Retained     |
|         | TOCRE1        | Initialized | Initialized         | Retained     |
|         | TIERE1        | Initialized | Initialized         | Retained     |
|         | RLDCRE1       | Initialized | Initialized         | Retained     |
|         | TSRE1         | Initialized | Initialized         | Retained     |
|         | PSCRE1        | Initialized | Initialized         | Retained     |
|         | SSTRE1        | Initialized | Initialized         | Retained     |
|         | CYLRE10       | Initialized | Initialized         | Retained     |
|         | CYLRE11       | Initialized | Initialized         | Retained     |
|         | CYLRE12       | Initialized | Initialized         | Retained     |
|         | CYLRE13       | Initialized | Initialized         | Retained     |
|         | DTRE10        | Initialized | Initialized         | Retained     |
|         | DTRE11        | Initialized | Initialized         | Retained     |
|         | DTRE12        | Initialized | Initialized         | Retained     |
|         | DTRE13        | Initialized | Initialized         | Retained     |
|         | CRLDE10       | Initialized | Initialized         | Retained     |
|         | CRLDE11       | Initialized | Initialized         | Retained     |
|         | CRLDE12       | Initialized | Initialized         | Retained     |
|         | CRLDE13       | Initialized | Initialized         | Retained     |
|         | DRLDE10       | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | DRLDE11       | Initialized | Initialized         | Retained     |
|         | DRLDE12       | Initialized | Initialized         | Retained     |
|         | DRLDE13       | Initialized | Initialized         | Retained     |
|         | TCNTE10       | Initialized | Initialized         | Retained     |
|         | TCNTE11       | Initialized | Initialized         | Retained     |
|         | TCNTE12       | Initialized | Initialized         | Retained     |
|         | TCNTE13       | Initialized | Initialized         | Retained     |
|         | TCRE2         | Initialized | Initialized         | Retained     |
|         | TOCRE2        | Initialized | Initialized         | Retained     |
|         | TIERE2        | Initialized | Initialized         | Retained     |
|         | RLDCRE2       | Initialized | Initialized         | Retained     |
|         | TSRE2         | Initialized | Initialized         | Retained     |
|         | PSCRE2        | Initialized | Initialized         | Retained     |
|         | SSTRE2        | Initialized | Initialized         | Retained     |
|         | CYLRE20       | Initialized | Initialized         | Retained     |
|         | CYLRE21       | Initialized | Initialized         | Retained     |
|         | CYLRE22       | Initialized | Initialized         | Retained     |
|         | CYLRE23       | Initialized | Initialized         | Retained     |
|         | DTRE20        | Initialized | Initialized         | Retained     |
|         | DTRE21        | Initialized | Initialized         | Retained     |
|         | DTRE22        | Initialized | Initialized         | Retained     |
|         | DTRE23        | Initialized | Initialized         | Retained     |
|         | CRLDE20       | Initialized | Initialized         | Retained     |
|         | CRLDE21       | Initialized | Initialized         | Retained     |
|         | CRLDE22       | Initialized | Initialized         | Retained     |
|         | CRLDE23       | Initialized | Initialized         | Retained     |
|         | DRLDE20       | Initialized | Initialized         | Retained     |
|         | DRLDE21       | Initialized | Initialized         | Retained     |
|         | DRLDE22       | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | DRLDE23       | Initialized | Initialized         | Retained     |
|         | TCNTE20       | Initialized | Initialized         | Retained     |
|         | TCNTE21       | Initialized | Initialized         | Retained     |
|         | TCNTE22       | Initialized | Initialized         | Retained     |
|         | TCNTE23       | Initialized | Initialized         | Retained     |
|         | TCRE3         | Initialized | Initialized         | Retained     |
|         | TOCRE3        | Initialized | Initialized         | Retained     |
|         | TIERE3        | Initialized | Initialized         | Retained     |
|         | RLDCRE3       | Initialized | Initialized         | Retained     |
|         | TSRE3         | Initialized | Initialized         | Retained     |
|         | PSCRE3        | Initialized | Initialized         | Retained     |
|         | SSTRE3        | Initialized | Initialized         | Retained     |
|         | CYLRE30       | Initialized | Initialized         | Retained     |
|         | CYLRE31       | Initialized | Initialized         | Retained     |
|         | CYLRE32       | Initialized | Initialized         | Retained     |
|         | CYLRE33       | Initialized | Initialized         | Retained     |
|         | DTRE30        | Initialized | Initialized         | Retained     |
|         | DTRE31        | Initialized | Initialized         | Retained     |
|         | DTRE32        | Initialized | Initialized         | Retained     |
|         | DTRE33        | Initialized | Initialized         | Retained     |
|         | CRLDE30       | Initialized | Initialized         | Retained     |
|         | CRLDE31       | Initialized | Initialized         | Retained     |
|         | CRLDE32       | Initialized | Initialized         | Retained     |
|         | CRLDE33       | Initialized | Initialized         | Retained     |
|         | DRLDE30       | Initialized | Initialized         | Retained     |
|         | DRLDE31       | Initialized | Initialized         | Retained     |
|         | DRLDE32       | Initialized | Initialized         | Retained     |
|         | DRLDE33       | Initialized | Initialized         | Retained     |
|         | TCNTE30       | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | TCNTE31       | Initialized | Initialized         | Retained     |
|         | TCNTE32       | Initialized | Initialized         | Retained     |
|         | TCNTE33       | Initialized | Initialized         | Retained     |
|         | TCRE4         | Initialized | Initialized         | Retained     |
|         | TOCRE4        | Initialized | Initialized         | Retained     |
|         | TIERE4        | Initialized | Initialized         | Retained     |
|         | RLDCRE4       | Initialized | Initialized         | Retained     |
|         | TSRE4         | Initialized | Initialized         | Retained     |
|         | PSCRE4        | Initialized | Initialized         | Retained     |
|         | SSTRE4        | Initialized | Initialized         | Retained     |
|         | CYLRE40       | Initialized | Initialized         | Retained     |
|         | CYLRE41       | Initialized | Initialized         | Retained     |
|         | CYLRE42       | Initialized | Initialized         | Retained     |
|         | CYLRE43       | Initialized | Initialized         | Retained     |
|         | DTRE40        | Initialized | Initialized         | Retained     |
|         | DTRE41        | Initialized | Initialized         | Retained     |
|         | DTRE42        | Initialized | Initialized         | Retained     |
|         | DTRE43        | Initialized | Initialized         | Retained     |
|         | CRLDE40       | Initialized | Initialized         | Retained     |
|         | CRLDE41       | Initialized | Initialized         | Retained     |
|         | CRLDE42       | Initialized | Initialized         | Retained     |
|         | CRLDE43       | Initialized | Initialized         | Retained     |
|         | DRLDE40       | Initialized | Initialized         | Retained     |
|         | DRLDE41       | Initialized | Initialized         | Retained     |
|         | DRLDE42       | Initialized | Initialized         | Retained     |
|         | DRLDE43       | Initialized | Initialized         | Retained     |
|         | TCNTE40       | Initialized | Initialized         | Retained     |
|         | TCNTE41       | Initialized | Initialized         | Retained     |
|         | TCNTE42       | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | TCNTE43       | Initialized | Initialized         | Retained     |
|         | TCRE5         | Initialized | Initialized         | Retained     |
|         | TOCRE5        | Initialized | Initialized         | Retained     |
|         | TIERE5        | Initialized | Initialized         | Retained     |
|         | RLDCRE5       | Initialized | Initialized         | Retained     |
|         | TSRE5         | Initialized | Initialized         | Retained     |
|         | PSCRE5        | Initialized | Initialized         | Retained     |
|         | SSTRE5        | Initialized | Initialized         | Retained     |
|         | CYLRE50       | Initialized | Initialized         | Retained     |
|         | CYLRE51       | Initialized | Initialized         | Retained     |
|         | CYLRE52       | Initialized | Initialized         | Retained     |
|         | CYLRE53       | Initialized | Initialized         | Retained     |
|         | DTRE50        | Initialized | Initialized         | Retained     |
|         | DTRE51        | Initialized | Initialized         | Retained     |
|         | DTRE52        | Initialized | Initialized         | Retained     |
|         | DTRE53        | Initialized | Initialized         | Retained     |
|         | CRLDE50       | Initialized | Initialized         | Retained     |
|         | CRLDE51       | Initialized | Initialized         | Retained     |
|         | CRLDE52       | Initialized | Initialized         | Retained     |
|         | CRLDE53       | Initialized | Initialized         | Retained     |
|         | DRLDE50       | Initialized | Initialized         | Retained     |
|         | DRLDE51       | Initialized | Initialized         | Retained     |
|         | DRLDE52       | Initialized | Initialized         | Retained     |
|         | DRLDE53       | Initialized | Initialized         | Retained     |
|         | TCNTE50       | Initialized | Initialized         | Retained     |
|         | TCNTE51       | Initialized | Initialized         | Retained     |
|         | TCNTE52       | Initialized | Initialized         | Retained     |
|         | TCNTE53       | Initialized | Initialized         | Retained     |
|         | TSTRF         | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |  |
|---------|---------------|-------------|---------------------|--------------|--|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |  |
| ATU-III | NCCRF         | Initialized | Initialized         | Retained     |  |
|         | NCNTFA0       | Initialized | Initialized         | Retained     |  |
|         | NCRFA0        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA1       | Initialized | Initialized         | Retained     |  |
|         | NCRFA1        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA2       | Initialized | Initialized         | Retained     |  |
|         | NCRFA2        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA3       | Initialized | Initialized         | Retained     |  |
|         | NCRFA3        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA4       | Initialized | Initialized         | Retained     |  |
|         | NCRFA4        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA5       | Initialized | Initialized         | Retained     |  |
|         | NCRFA5        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA6       | Initialized | Initialized         | Retained     |  |
|         | NCRFA6        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA7       | Initialized | Initialized         | Retained     |  |
|         | NCRFA7        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA8       | Initialized | Initialized         | Retained     |  |
|         | NCRFA8        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA9       | Initialized | Initialized         | Retained     |  |
|         | NCRFA9        | Initialized | Initialized         | Retained     |  |
|         | NCNTFA10      | Initialized | Initialized         | Retained     |  |
|         | NCRFA10       | Initialized | Initialized         | Retained     |  |
|         | NCNTFA11      | Initialized | Initialized         | Retained     |  |
|         | NCRFA11       | Initialized | Initialized         | Retained     |  |
|         | NCNTFA12      | Initialized | Initialized         | Retained     |  |
|         | NCRFA12       | Initialized | Initialized         | Retained     |  |
|         | NCNTFA13      | Initialized | Initialized         | Retained     |  |
|         | NCRFA13       | Initialized | Initialized         | Retained     |  |
|         |               |             |                     |              |  |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | NCNTFA14      | Initialized | Initialized         | Retained     |
|         | NCRFA14       | Initialized | Initialized         | Retained     |
|         | NCNTFA15      | Initialized | Initialized         | Retained     |
|         | NCRFA15       | Initialized | Initialized         | Retained     |
|         | NCNTFA16      | Initialized | Initialized         | Retained     |
|         | NCRFA16       | Initialized | Initialized         | Retained     |
|         | NCNTFA17      | Initialized | Initialized         | Retained     |
|         | NCRFA17       | Initialized | Initialized         | Retained     |
|         | NCNTFA18      | Initialized | Initialized         | Retained     |
|         | NCRFA18       | Initialized | Initialized         | Retained     |
|         | NCNTFA19      | Initialized | Initialized         | Retained     |
|         | NCRFA19       | Initialized | Initialized         | Retained     |
|         | NCNTFB0       | Initialized | Initialized         | Retained     |
|         | NCRFB0        | Initialized | Initialized         | Retained     |
|         | NCNTFB1       | Initialized | Initialized         | Retained     |
|         | NCRFB1        | Initialized | Initialized         | Retained     |
|         | NCNTFB2       | Initialized | Initialized         | Retained     |
|         | NCRFB2        | Initialized | Initialized         | Retained     |
|         | TCRF0         | Initialized | Initialized         | Retained     |
|         | TIERF0        | Initialized | Initialized         | Retained     |
|         | TSRF0         | Initialized | Initialized         | Retained     |
|         | ECNTAF0       | Initialized | Initialized         | Retained     |
|         | ECNTBF0       | Initialized | Initialized         | Retained     |
|         | GRBF0         | Initialized | Initialized         | Retained     |
|         | ECNTCF0       | Initialized | Initialized         | Retained     |
|         | GRAF0         | Initialized | Initialized         | Retained     |
|         | CDRF0         | Initialized | Initialized         | Retained     |
|         | GRCF0         | Initialized | Initialized         | Retained     |
|         | TCRF1         | Initialized | Initialized         | Retained     |
|         |               |             |                     |              |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | TIERF1        | Initialized | Initialized         | Retained     |
|         | TSRF1         | Initialized | Initialized         | Retained     |
|         | ECNTAF1       | Initialized | Initialized         | Retained     |
|         | ECNTBF1       | Initialized | Initialized         | Retained     |
|         | GRBF1         | Initialized | Initialized         | Retained     |
|         | ECNTCF1       | Initialized | Initialized         | Retained     |
|         | GRAF1         | Initialized | Initialized         | Retained     |
|         | CDRF1         | Initialized | Initialized         | Retained     |
|         | GRCF1         | Initialized | Initialized         | Retained     |
|         | TCRF2         | Initialized | Initialized         | Retained     |
|         | TIERF2        | Initialized | Initialized         | Retained     |
|         | TSRF2         | Initialized | Initialized         | Retained     |
|         | ECNTAF2       | Initialized | Initialized         | Retained     |
|         | ECNTBF2       | Initialized | Initialized         | Retained     |
|         | GRBF2         | Initialized | Initialized         | Retained     |
|         | ECNTCF2       | Initialized | Initialized         | Retained     |
|         | GRAF2         | Initialized | Initialized         | Retained     |
|         | CDRF2         | Initialized | Initialized         | Retained     |
|         | GRCF2         | Initialized | Initialized         | Retained     |
|         | TCRF3         | Initialized | Initialized         | Retained     |
|         | TIERF3        | Initialized | Initialized         | Retained     |
|         | TSRF3         | Initialized | Initialized         | Retained     |
|         | ECNTAF3       | Initialized | Initialized         | Retained     |
|         | ECNTBF3       | Initialized | Initialized         | Retained     |
|         | GRBF3         | Initialized | Initialized         | Retained     |
|         | ECNTCF3       | Initialized | Initialized         | Retained     |
|         | GRAF3         | Initialized | Initialized         | Retained     |
|         | CDRF3         | Initialized | Initialized         | Retained     |
|         | GRCF3         | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | TCRF4         | Initialized | Initialized         | Retained     |
|         | TIERF4        | Initialized | Initialized         | Retained     |
|         | TSRF4         | Initialized | Initialized         | Retained     |
|         | ECNTAF4       | Initialized | Initialized         | Retained     |
|         | ECNTBF4       | Initialized | Initialized         | Retained     |
|         | GRBF4         | Initialized | Initialized         | Retained     |
|         | ECNTCF4       | Initialized | Initialized         | Retained     |
|         | GRAF4         | Initialized | Initialized         | Retained     |
|         | CDRF4         | Initialized | Initialized         | Retained     |
|         | GRCF4         | Initialized | Initialized         | Retained     |
|         | TCRF5         | Initialized | Initialized         | Retained     |
|         | TIERF5        | Initialized | Initialized         | Retained     |
|         | TSRF5         | Initialized | Initialized         | Retained     |
|         | ECNTAF5       | Initialized | Initialized         | Retained     |
|         | ECNTBF5       | Initialized | Initialized         | Retained     |
|         | GRBF5         | Initialized | Initialized         | Retained     |
|         | ECNTCF5       | Initialized | Initialized         | Retained     |
|         | GRAF5         | Initialized | Initialized         | Retained     |
|         | CDRF5         | Initialized | Initialized         | Retained     |
|         | GRCF5         | Initialized | Initialized         | Retained     |
|         | TCRF6         | Initialized | Initialized         | Retained     |
|         | TIERF6        | Initialized | Initialized         | Retained     |
|         | TSRF6         | Initialized | Initialized         | Retained     |
|         | ECNTAF6       | Initialized | Initialized         | Retained     |
|         | ECNTBF6       | Initialized | Initialized         | Retained     |
|         | GRBF6         | Initialized | Initialized         | Retained     |
|         | ECNTCF6       | Initialized | Initialized         | Retained     |
|         | GRAF6         | Initialized | Initialized         | Retained     |
|         | CDRF6         | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | GRCF6         | Initialized | Initialized         | Retained     |
|         | TCRF7         | Initialized | Initialized         | Retained     |
|         | TIERF7        | Initialized | Initialized         | Retained     |
|         | TSRF7         | Initialized | Initialized         | Retained     |
|         | ECNTAF7       | Initialized | Initialized         | Retained     |
|         | ECNTBF7       | Initialized | Initialized         | Retained     |
|         | GRBF7         | Initialized | Initialized         | Retained     |
|         | ECNTCF7       | Initialized | Initialized         | Retained     |
|         | GRAF7         | Initialized | Initialized         | Retained     |
|         | CDRF7         | Initialized | Initialized         | Retained     |
|         | GRCF7         | Initialized | Initialized         | Retained     |
|         | TCRF8         | Initialized | Initialized         | Retained     |
|         | TIERF8        | Initialized | Initialized         | Retained     |
|         | TSRF8         | Initialized | Initialized         | Retained     |
|         | ECNTAF8       | Initialized | Initialized         | Retained     |
|         | ECNTBF8       | Initialized | Initialized         | Retained     |
|         | GRBF8         | Initialized | Initialized         | Retained     |
|         | ECNTCF8       | Initialized | Initialized         | Retained     |
|         | GRAF8         | Initialized | Initialized         | Retained     |
|         | CDRF8         | Initialized | Initialized         | Retained     |
|         | GRCF8         | Initialized | Initialized         | Retained     |
|         | TCRF9         | Initialized | Initialized         | Retained     |
|         | TIERF9        | Initialized | Initialized         | Retained     |
|         | TSRF9         | Initialized | Initialized         | Retained     |
|         | ECNTAF9       | Initialized | Initialized         | Retained     |
|         | ECNTBF9       | Initialized | Initialized         | Retained     |
|         | GRBF9         | Initialized | Initialized         | Retained     |
|         | ECNTCF9       | Initialized | Initialized         | Retained     |
|         | GRAF9         | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | CDRF9         | Initialized | Initialized         | Retained     |
|         | GRCF9         | Initialized | Initialized         | Retained     |
|         | TCRF10        | Initialized | Initialized         | Retained     |
|         | TIERF10       | Initialized | Initialized         | Retained     |
|         | TSRF10        | Initialized | Initialized         | Retained     |
|         | ECNTAF10      | Initialized | Initialized         | Retained     |
|         | ECNTBF10      | Initialized | Initialized         | Retained     |
|         | GRBF10        | Initialized | Initialized         | Retained     |
|         | ECNTCF10      | Initialized | Initialized         | Retained     |
|         | GRAF10        | Initialized | Initialized         | Retained     |
|         | CDRF10        | Initialized | Initialized         | Retained     |
|         | GRCF10        | Initialized | Initialized         | Retained     |
|         | TCRF11        | Initialized | Initialized         | Retained     |
|         | TIERF11       | Initialized | Initialized         | Retained     |
|         | TSRF11        | Initialized | Initialized         | Retained     |
|         | ECNTAF11      | Initialized | Initialized         | Retained     |
|         | ECNTBF11      | Initialized | Initialized         | Retained     |
|         | GRBF11        | Initialized | Initialized         | Retained     |
|         | ECNTCF11      | Initialized | Initialized         | Retained     |
|         | GRAF11        | Initialized | Initialized         | Retained     |
|         | CDRF11        | Initialized | Initialized         | Retained     |
|         | GRCF11        | Initialized | Initialized         | Retained     |
|         | TCRF12        | Initialized | Initialized         | Retained     |
|         | TIERF12       | Initialized | Initialized         | Retained     |
|         | TSRF12        | Initialized | Initialized         | Retained     |
|         | ECNTAF12      | Initialized | Initialized         | Retained     |
|         | ECNTBF12      | Initialized | Initialized         | Retained     |
|         | GRBF12        | Initialized | Initialized         | Retained     |
|         | ECNTCF12      | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | GRAF12        | Initialized | Initialized         | Retained     |
|         | CDRF12        | Initialized | Initialized         | Retained     |
|         | GRCF12        | Initialized | Initialized         | Retained     |
|         | GRDF12        | Initialized | Initialized         | Retained     |
|         | TCRF13        | Initialized | Initialized         | Retained     |
|         | TIERF13       | Initialized | Initialized         | Retained     |
|         | TSRF13        | Initialized | Initialized         | Retained     |
|         | ECNTAF13      | Initialized | Initialized         | Retained     |
|         | ECNTBF13      | Initialized | Initialized         | Retained     |
|         | GRBF13        | Initialized | Initialized         | Retained     |
|         | ECNTCF13      | Initialized | Initialized         | Retained     |
|         | GRAF13        | Initialized | Initialized         | Retained     |
|         | CDRF13        | Initialized | Initialized         | Retained     |
|         | GRCF13        | Initialized | Initialized         | Retained     |
|         | GRDF13        | Initialized | Initialized         | Retained     |
|         | TCRF14        | Initialized | Initialized         | Retained     |
|         | TIERF14       | Initialized | Initialized         | Retained     |
|         | TSRF14        | Initialized | Initialized         | Retained     |
|         | ECNTAF14      | Initialized | Initialized         | Retained     |
|         | ECNTBF14      | Initialized | Initialized         | Retained     |
|         | GRBF14        | Initialized | Initialized         | Retained     |
|         | ECNTCF14      | Initialized | Initialized         | Retained     |
|         | GRAF14        | Initialized | Initialized         | Retained     |
|         | CDRF14        | Initialized | Initialized         | Retained     |
|         | GRCF14        | Initialized | Initialized         | Retained     |
|         | GRDF14        | Initialized | Initialized         | Retained     |
|         | TCRF15        | Initialized | Initialized         | Retained     |
|         | TIERF15       | Initialized | Initialized         | Retained     |
|         | TSRF15        | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | ower-Down State |  |
|---------|---------------|-------------|---------------------|-----------------|--|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep           |  |
| ATU-III | ECNTAF15      | Initialized | Initialized         | Retained        |  |
|         | ECNTBF15      | Initialized | Initialized         | Retained        |  |
|         | GRBF15        | Initialized | Initialized         | Retained        |  |
|         | ECNTCF15      | Initialized | Initialized         | Retained        |  |
|         | GRAF15        | Initialized | Initialized         | Retained        |  |
|         | CDRF15        | Initialized | Initialized         | Retained        |  |
|         | GRCF15        | Initialized | Initialized         | Retained        |  |
|         | GRDF15        | Initialized | Initialized         | Retained        |  |
|         | TCRF16        | Initialized | Initialized         | Retained        |  |
|         | TIERF16       | Initialized | Initialized         | Retained        |  |
|         | TSRF16        | Initialized | Initialized         | Retained        |  |
|         | ECNTAF16      | Initialized | Initialized         | Retained        |  |
|         | ECNTBF16      | Initialized | Initialized         | Retained        |  |
|         | GRBF16        | Initialized | Initialized         | Retained        |  |
|         | ECNTCF16      | Initialized | Initialized         | Retained        |  |
|         | GRAF16        | Initialized | Initialized         | Retained        |  |
|         | CDRF16        | Initialized | Initialized         | Retained        |  |
|         | GRCF16        | Initialized | Initialized         | Retained        |  |
|         | TCRF17        | Initialized | Initialized         | Retained        |  |
|         | TIERF17       | Initialized | Initialized         | Retained        |  |
|         | TSRF17        | Initialized | Initialized         | Retained        |  |
|         | ECNTAF17      | Initialized | Initialized         | Retained        |  |
|         | ECNTBF17      | Initialized | Initialized         | Retained        |  |
|         | GRBF17        | Initialized | Initialized         | Retained        |  |
|         | ECNTCF17      | Initialized | Initialized         | Retained        |  |
|         | GRAF17        | Initialized | Initialized         | Retained        |  |
|         | CDRF17        | Initialized | Initialized         | Retained        |  |
|         | GRCF17        | Initialized | Initialized         | Retained        |  |
|         | TCRF18        | Initialized | Initialized         | Retained        |  |

|         |               | Reset State | Power-D             | Down State |
|---------|---------------|-------------|---------------------|------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep      |
| ATU-III | TIERF18       | Initialized | Initialized         | Retained   |
|         | TSRF18        | Initialized | Initialized         | Retained   |
|         | ECNTAF18      | Initialized | Initialized         | Retained   |
|         | ECNTBF18      | Initialized | Initialized         | Retained   |
|         | GRBF18        | Initialized | Initialized         | Retained   |
|         | ECNTCF18      | Initialized | Initialized         | Retained   |
|         | GRAF18        | Initialized | Initialized         | Retained   |
|         | CDRF18        | Initialized | Initialized         | Retained   |
|         | GRCF18        | Initialized | Initialized         | Retained   |
|         | TCRF19        | Initialized | Initialized         | Retained   |
|         | TIERF19       | Initialized | Initialized         | Retained   |
|         | TSRF19        | Initialized | Initialized         | Retained   |
|         | ECNTAF19      | Initialized | Initialized         | Retained   |
|         | ECNTBF19      | Initialized | Initialized         | Retained   |
|         | GRBF19        | Initialized | Initialized         | Retained   |
|         | ECNTCF19      | Initialized | Initialized         | Retained   |
|         | GRAF19        | Initialized | Initialized         | Retained   |
|         | CDRF19        | Initialized | Initialized         | Retained   |
|         | GRCF19        | Initialized | Initialized         | Retained   |
|         | TSTRG         | Initialized | Initialized         | Retained   |
|         | TCRG0         | Initialized | Initialized         | Retained   |
|         | TSRG0         | Initialized | Initialized         | Retained   |
|         | TCNTG0        | Initialized | Initialized         | Retained   |
|         | OCRG0         | Initialized | Initialized         | Retained   |
|         | TCRG1         | Initialized | Initialized         | Retained   |
|         | TSRG1         | Initialized | Initialized         | Retained   |
|         | TCNTG1        | Initialized | Initialized         | Retained   |
|         | OCRG1         | Initialized | Initialized         | Retained   |
|         | TCRG2         | Initialized | Initialized         | Retained   |

|         | Register Name | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  |               | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | TSRG2         | Initialized | Initialized         | Retained     |
|         | TCNTG2        | Initialized | Initialized         | Retained     |
|         | OCRG2         | Initialized | Initialized         | Retained     |
|         | TCRG3         | Initialized | Initialized         | Retained     |
|         | TSRG3         | Initialized | Initialized         | Retained     |
|         | TCNTG3        | Initialized | Initialized         | Retained     |
|         | OCRG3         | Initialized | Initialized         | Retained     |
|         | TCRG4         | Initialized | Initialized         | Retained     |
|         | TSRG4         | Initialized | Initialized         | Retained     |
|         | TCNTG4        | Initialized | Initialized         | Retained     |
|         | OCRG4         | Initialized | Initialized         | Retained     |
|         | TCRG5         | Initialized | Initialized         | Retained     |
|         | TSRG5         | Initialized | Initialized         | Retained     |
|         | TCNTG5        | Initialized | Initialized         | Retained     |
|         | OCRG5         | Initialized | Initialized         | Retained     |
|         | TCRH          | Initialized | Initialized         | Retained     |
|         | TSRH          | Initialized | Initialized         | Retained     |
|         | TCNT1H        | Initialized | Initialized         | Retained     |
|         | OCR1H         | Initialized | Initialized         | Retained     |
|         | TCNT2H        | Initialized | Initialized         | Retained     |
|         | TSTRJ         | Initialized | Initialized         | Retained     |
|         | TCRJ0         | Initialized | Initialized         | Retained     |
|         | FCRJ0         | Initialized | Initialized         | Retained     |
|         | TSRJ0         | Initialized | Initialized         | Retained     |
|         | TIERJ0        | Initialized | Initialized         | Retained     |
|         | FDNRJ0        | Initialized | Initialized         | Retained     |
|         | NCNTJ0        | Initialized | Initialized         | Retained     |
|         | NCRJ0         | Initialized | Initialized         | Retained     |
|         | TCNTJ0        | Initialized | Initialized         | Retained     |

|         |               | Reset State | Powe                | r-Down State |
|---------|---------------|-------------|---------------------|--------------|
| Module  | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ATU-III | OCRJ0         | Initialized | Initialized         | Retained     |
|         | FIFOJ0        | Initialized | Initialized         | Retained     |
|         | TCRJ1         | Initialized | Initialized         | Retained     |
|         | FCRJ1         | Initialized | Initialized         | Retained     |
|         | TSRJ1         | Initialized | Initialized         | Retained     |
|         | TIERJ1        | Initialized | Initialized         | Retained     |
|         | FDNRJ1        | Initialized | Initialized         | Retained     |
|         | NCNTJ1        | Initialized | Initialized         | Retained     |
|         | NCRJ1         | Initialized | Initialized         | Retained     |
|         | TCNTJ1        | Initialized | Initialized         | Retained     |
|         | OCRJ1         | Initialized | Initialized         | Retained     |
|         | FIFOJ1        | Initialized | Initialized         | Retained     |
| WDT     | WTCR          | Initialized | Initialized         | Retained     |
|         | WTCNT         | Initialized | Initialized         | Retained     |
|         | WTSR          | Initialized | Initialized         | Retained     |
|         | WRCR          | Initialized | Initialized         | Retained     |
| CMT     | CMSTR         | Initialized | Initialized         | Retained     |
|         | CMCR_0        | Initialized | Initialized         | Retained     |
|         | CMSR_0        | Initialized | Initialized         | Retained     |
|         | CMCNT_0       | Initialized | Initialized         | Retained     |
|         | CMCOR_0       | Initialized | Initialized         | Retained     |
|         | CMCR_1        | Initialized | Initialized         | Retained     |
|         | CMSR_1        | Initialized | Initialized         | Retained     |
|         | CMCNT_1       | Initialized | Initialized         | Retained     |
|         | CMCOR_1       | Initialized | Initialized         | Retained     |
| SCI     | SCSMR1A       | Initialized | Initialized         | Retained     |
|         | SCBRR1A       | Initialized | Initialized         | Retained     |
|         | SCSCR1A       | Initialized | Initialized         | Retained     |
|         | SCTDR1A       | Initialized | Initialized         | Retained     |

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| SCI    | SCSSR1A       | Initialized | Initialized         | Retained     |
|        | SCRDR1A       | Initialized | Initialized         | Retained     |
|        | SCSMR1B       | Initialized | Initialized         | Retained     |
|        | SCBRR1B       | Initialized | Initialized         | Retained     |
|        | SCSCR1B       | Initialized | Initialized         | Retained     |
|        | SCTDR1B       | Initialized | Initialized         | Retained     |
|        | SCSSR1B       | Initialized | Initialized         | Retained     |
|        | SCRDR1B       | Initialized | Initialized         | Retained     |
|        | SCSMR1C       | Initialized | Initialized         | Retained     |
|        | SCBRR1C       | Initialized | Initialized         | Retained     |
|        | SCSCR1C       | Initialized | Initialized         | Retained     |
|        | SCTDR1C       | Initialized | Initialized         | Retained     |
|        | SCSSR1C       | Initialized | Initialized         | Retained     |
|        | SCRDR1C       | Initialized | Initialized         | Retained     |
|        | SCSMR1D       | Initialized | Initialized         | Retained     |
|        | SCBRR1D       | Initialized | Initialized         | Retained     |
|        | SCSCR1D       | Initialized | Initialized         | Retained     |
|        | SCTDR1D       | Initialized | Initialized         | Retained     |
|        | SCSSR1D       | Initialized | Initialized         | Retained     |
|        | SCRDR1D       | Initialized | Initialized         | Retained     |
|        | SCSMR1E       | Initialized | Initialized         | Retained     |
|        | SCBRR1E       | Initialized | Initialized         | Retained     |
|        | SCSCR1E       | Initialized | Initialized         | Retained     |
|        | SCTDR1E       | Initialized | Initialized         | Retained     |
|        | SCSSR1E       | Initialized | Initialized         | Retained     |
|        | SCRDR1E       | Initialized | Initialized         | Retained     |
| RSPI   | SPCRA         | Initialized | Initialized         | Retained     |
|        | SSLPA         | Initialized | Initialized         | Retained     |
|        | SPPCRA        | Initialized | Initialized         | Retained     |

|        |               | Reset State | Powe                | r-Down State |
|--------|---------------|-------------|---------------------|--------------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| RSPI   | SPSRA         | Initialized | Initialized         | Retained     |
|        | SPDRA         | Initialized | Initialized         | Retained     |
|        | SPSCRA        | Initialized | Initialized         | Retained     |
|        | SPSSRA        | Initialized | Initialized         | Retained     |
|        | SPBRA         | Initialized | Initialized         | Retained     |
|        | SPCKDA        | Initialized | Initialized         | Retained     |
|        | SSLNDA        | Initialized | Initialized         | Retained     |
|        | SPNDA         | Initialized | Initialized         | Retained     |
|        | SPCMDA0       | Initialized | Initialized         | Retained     |
|        | SPCMDA1       | Initialized | Initialized         | Retained     |
|        | SPCMDA2       | Initialized | Initialized         | Retained     |
|        | SPCMDA3       | Initialized | Initialized         | Retained     |
|        | SPCMDA4       | Initialized | Initialized         | Retained     |
|        | SPCMDA5       | Initialized | Initialized         | Retained     |
|        | SPCMDA6       | Initialized | Initialized         | Retained     |
|        | SPCMDA7       | Initialized | Initialized         | Retained     |
|        | SPCRB         | Initialized | Initialized         | Retained     |
|        | SSLPB         | Initialized | Initialized         | Retained     |
|        | SPPCRB        | Initialized | Initialized         | Retained     |
|        | SPSRB         | Initialized | Initialized         | Retained     |
|        | SPDRB         | Initialized | Initialized         | Retained     |
|        | SPSCRB        | Initialized | Initialized         | Retained     |
|        | SPSSRB        | Initialized | Initialized         | Retained     |
|        | SPBRB         | Initialized | Initialized         | Retained     |
|        | SPCKDB        | Initialized | Initialized         | Retained     |
|        | SSLNDB        | Initialized | Initialized         | Retained     |
|        | SPNDB         | Initialized | Initialized         | Retained     |
|        | SPCMDB0       | Initialized | Initialized         | Retained     |
|        | SPCMDB1       | Initialized | Initialized         | Retained     |

| Register Name  |          |               | Reset State | Power-Down State |          |
|--|----------|---------------|-------------|------------------|----------|
| SPCMDB3 Initialized Initialized Retained SPCMDB4 Initialized Initialized Retained SPCMDB5 Initialized Initialized Retained SPCMDB6 Initialized Initialized Retained SPCMDB7 Initialized Initialized Retained SPCMDB7 Initialized Initialized Retained SPCRC Initialized Initialized Retained SPCRC Initialized Initialized Retained SPCRC Initialized Initialized Retained SPPCRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPSCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained SPCMDC8 Initialized Initialized Retained SPCMDC9 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained SPCMDC9 Initialized Initialized Retained SPCMDC9 Initialized Initialized Retained SPCMDC9 Initialized Retained SP | Module   | Register Name | Power-On    |                  | Sleep    |
| SPCMDB4 Initialized Initialized Retained SPCMDB5 Initialized Initialized Retained SPCMDB6 Initialized Initialized Retained SPCMDB7 Initialized Initialized Retained SPCRC Initialized Initialized Retained SPCRC Initialized Initialized Retained SSLPC Initialized Initialized Retained SPCRC Initialized Initialized Retained SPCRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCMDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained   | RSPI     | SPCMDB2       | Initialized | Initialized      | Retained |
| SPCMDB5 Initialized Initialized Retained SPCMDB6 Initialized Initialized Retained SPCMDB7 Initialized Initialized Retained SPCRC Initialized Initialized Retained SSLPC Initialized Initialized Retained SPCRC Initialized Initialized Retained SPPCRC Initialized Initialized Retained SPPCRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPBRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained  |          | SPCMDB3       | Initialized | Initialized      | Retained |
| SPCMDB6 Initialized Initialized Retained SPCMDB7 Initialized Initialized Retained SPCRC Initialized Initialized Retained SSLPC Initialized Initialized Retained SPCRC Initialized Initialized Retained SPPCRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPDRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCMDC Initialized Initialized Retained SPCMDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 (RCAN_A) GSR Initialized Initialized Retained RCAN-TL1 (RCAN_A) Entained Initialized Initialized Retained RCAN-TL1 Initialized Initialized Retained  |          | SPCMDB4       | Initialized | Initialized      | Retained |
| SPCMDB7 Initialized Initialized Retained SPCRC Initialized Initialized Retained SSLPC Initialized Initialized Retained SPPCRC Initialized Initialized Retained SPPCRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPDRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCMDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 (RCAN_A)  RCAN-TL1 (RCAN_A) GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained   |          | SPCMDB5       | Initialized | Initialized      | Retained |
| SPCRC Initialized Initialized Retained SSLPC Initialized Initialized Retained SPPCRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained BCR1 Initialized Initialized Retained   |          | SPCMDB6       | Initialized | Initialized      | Retained |
| SSLPC Initialized Initialized Retained SPPCRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPDRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPBRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 (RCAN_A) GSR Initialized Initialized Retained BCR1 Initialized Retained Retained   |          | SPCMDB7       | Initialized | Initialized      | Retained |
| SPPCRC Initialized Initialized Retained SPSRC Initialized Initialized Retained SPDRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPBRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained RCAN-TL1 Initialized Initialized Retained  |          | SPCRC         | Initialized | Initialized      | Retained |
| SPSRC Initialized Initialized Retained SPDRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPBRC Initialized Initialized Retained SPBRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SSLNDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained RCAN_A)  RCAN-TL1 MCR Initialized Initialized Retained RCAN_BCR1 Initialized Initialized Retained Retained   |          | SSLPC         | Initialized | Initialized      | Retained |
| SPDRC Initialized Initialized Retained SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPBRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SSLNDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained RCAN_A) BCR1 Initialized Initialized Retained  |          | SPPCRC        | Initialized | Initialized      | Retained |
| SPSCRC Initialized Initialized Retained SPSSRC Initialized Initialized Retained SPBRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SSLNDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained RCAN_A) GSR Initialized Initialized Retained Initialized Retained Retained  |          | SPSRC         | Initialized | Initialized      | Retained |
| SPSRC Initialized Initialized Retained SPBRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SSLNDC Initialized Initialized Retained SSLNDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 (RCAN_A) GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained Retained Retained Retained   |          | SPDRC         | Initialized | Initialized      | Retained |
| SPBRC Initialized Initialized Retained SPCKDC Initialized Initialized Retained SSLNDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 (RCAN_A) GSR Initialized Initialized Retained BCR1 Initialized Retained  |          | SPSCRC        | Initialized | Initialized      | Retained |
| SPCKDC Initialized Initialized Retained  SSLNDC Initialized Initialized Retained  SPNDC Initialized Initialized Retained  SPCMDC0 Initialized Initialized Retained  SPCMDC1 Initialized Initialized Retained  SPCMDC2 Initialized Initialized Retained  SPCMDC3 Initialized Initialized Retained  SPCMDC4 Initialized Initialized Retained  SPCMDC5 Initialized Initialized Retained  SPCMDC5 Initialized Initialized Retained  SPCMDC6 Initialized Initialized Retained  SPCMDC7 Initialized Initialized Retained  RCAN-TL1 (RCAN_A)  GSR Initialized Initialized Retained  BCR1 Initialized Initialized Retained  Retained   |          | SPSSRC        | Initialized | Initialized      | Retained |
| SSLNDC Initialized Initialized Retained SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 (RCAN_A) GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained Retained Retained Retained Retained Retained Retained Retained   |          | SPBRC         | Initialized | Initialized      | Retained |
| SPNDC Initialized Initialized Retained SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained Retained   |          | SPCKDC        | Initialized | Initialized      | Retained |
| SPCMDC0 Initialized Initialized Retained SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained RCAN_A) GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained   |          | SSLNDC        | Initialized | Initialized      | Retained |
| SPCMDC1 Initialized Initialized Retained SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained Retained Retained  |          | SPNDC         | Initialized | Initialized      | Retained |
| SPCMDC2 Initialized Initialized Retained SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained   |          | SPCMDC0       | Initialized | Initialized      | Retained |
| SPCMDC3 Initialized Initialized Retained SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained Retained   |          | SPCMDC1       | Initialized | Initialized      | Retained |
| SPCMDC4 Initialized Initialized Retained SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained RCAN-TL1 MCR Initialized Initialized Retained (RCAN_A) GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained  |          | SPCMDC2       | Initialized | Initialized      | Retained |
| SPCMDC5 Initialized Initialized Retained SPCMDC6 Initialized Initialized Retained SPCMDC7 Initialized Initialized Retained  RCAN-TL1 MCR Initialized Initialized Retained  GSR Initialized Initialized Retained  BCR1 Initialized Initialized Retained   |          | SPCMDC3       | Initialized | Initialized      | Retained |
| SPCMDC6 Initialized Initialized Retained  SPCMDC7 Initialized Initialized Retained  RCAN-TL1 (RCAN_A) GSR Initialized Initialized Retained  BCR1 Initialized Initialized Retained  Retained Retained   |          | SPCMDC4       | Initialized | Initialized      | Retained |
| SPCMDC7 Initialized Initialized Retained  RCAN-TL1 (RCAN_A) MCR Initialized Initialized Retained  GSR Initialized Initialized Retained  BCR1 Initialized Initialized Retained  |          | SPCMDC5       | Initialized | Initialized      | Retained |
| RCAN-TL1 (RCAN_A) MCR Initialized Initialized Retained GSR Initialized Initialized Retained BCR1 Initialized Initialized Retained  |          | SPCMDC6       | Initialized | Initialized      | Retained |
| (RCAN_A)     GSR     Initialized     Initialized     Retained       BCR1     Initialized     Initialized     Retained  |          | SPCMDC7       | Initialized | Initialized      | Retained |
| BCR1 Initialized Initialized Retained  | -        | MCR           | Initialized | Initialized      | Retained |
|  | (RCAN_A) | GSR           | Initialized | Initialized      | Retained |
| BCR0 Initialized Initialized Retained  |          | BCR1          | Initialized | Initialized      | Retained |
|  |          | BCR0          | Initialized | Initialized      | Retained |

|          |               | Reset State | Powe                | r-Down State |
|----------|---------------|-------------|---------------------|--------------|
| Module   | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | IRR           | Initialized | Initialized         | Retained     |
| (RCAN_A) | IMR           | Initialized | Initialized         | Retained     |
|          | TEC/REC       | Initialized | Initialized         | Retained     |
|          | TXPR1         | Initialized | Initialized         | Retained     |
|          | TXPR0         | Initialized | Initialized         | Retained     |
|          | TXCR1         | Initialized | Initialized         | Retained     |
|          | TXCR0         | Initialized | Initialized         | Retained     |
|          | TXACK1        | Initialized | Initialized         | Retained     |
|          | TXACK0        | Initialized | Initialized         | Retained     |
|          | ABACK1        | Initialized | Initialized         | Retained     |
|          | ABACK0        | Initialized | Initialized         | Retained     |
|          | RXPR1         | Initialized | Initialized         | Retained     |
|          | RXPR0         | Initialized | Initialized         | Retained     |
|          | RFPR1         | Initialized | Initialized         | Retained     |
|          | RFPR0         | Initialized | Initialized         | Retained     |
|          | MBIMR1        | Initialized | Initialized         | Retained     |
|          | MBIMR0        | Initialized | Initialized         | Retained     |
|          | UMSR1         | Initialized | Initialized         | Retained     |
|          | UMSR0         | Initialized | Initialized         | Retained     |
|          | TTCR0         | Initialized | Initialized         | Retained     |
|          | CMAX_TEW      | Initialized | Initialized         | Retained     |
|          | RFTROFF       | Initialized | Initialized         | Retained     |
|          | TSR           | Initialized | Initialized         | Retained     |
|          | CCR           | Initialized | Initialized         | Retained     |
|          | TCNTR         | Initialized | Initialized         | Retained     |
|          | CYCTR         | Initialized | Initialized         | Retained     |
|          | RFMK          | Initialized | Initialized         | Retained     |
|          | TCMR0         | Initialized | Initialized         | Retained     |
|          | TCMR1         | Initialized | Initialized         | Retained     |

|          | Register Name |            | Reset State | Powe                | r-Down State |
|----------|---------------|------------|-------------|---------------------|--------------|
| Module   |               |            | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | TCMR2         |            | Initialized | Initialized         | Retained     |
| (RCAN_A) | TTTSE         | L          | Initialized | Initialized         | Retained     |
|          | MB[0].        | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |               | LAFM0      | Undefined   | Undefined           | Retained     |
|          |               | LAFM1      | Undefined   | Undefined           | Retained     |
|          |               | DATA_01    | Undefined   | Undefined           | Retained     |
|          |               | DATA_23    | Undefined   | Undefined           | Retained     |
|          |               | DATA_45    | Undefined   | Undefined           | Retained     |
|          |               | DATA_67    | Undefined   | Undefined           | Retained     |
|          |               | CONTROL1   | Initialized | Initialized         | Retained     |
|          |               | TTT        | Initialized | Initialized         | Retained     |
|          | MB[1].        | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |               | LAFM0      | Undefined   | Undefined           | Retained     |
|          |               | LAFM1      | Undefined   | Undefined           | Retained     |
|          |               | DATA_01    | Undefined   | Undefined           | Retained     |
|          |               | DATA_23    | Undefined   | Undefined           | Retained     |
|          |               | DATA_45    | Undefined   | Undefined           | Retained     |
|          |               | DATA_67    | Undefined   | Undefined           | Retained     |
|          |               | CONTROL1   | Initialized | Initialized         | Retained     |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[2].        | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |               | LAFM0      | Undefined   | Undefined           | Retained     |
|          |               | LAFM1      | Undefined   | Undefined           | Retained     |
|          |               | DATA_01    | Undefined   | Undefined           | Retained     |
|          |               | DATA_23    | Undefined   | Undefined           | Retained     |
|          |               | DATA_45    | Undefined   | Undefined           | Retained     |

|          |               |            | Reset State | Powe                | r-Down State |
|----------|---------------|------------|-------------|---------------------|--------------|
| Module   | Register Name |            | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[2].        | DATA_67    | Undefined   | Undefined           | Retained     |
| (RCAN_A) |               | CONTROL1   | Initialized | Initialized         | Retained     |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[3].        | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |               | LAFM0      | Undefined   | Undefined           | Retained     |
|          |               | LAFM1      | Undefined   | Undefined           | Retained     |
|          |               | DATA_01    | Undefined   | Undefined           | Retained     |
|          |               | DATA_23    | Undefined   | Undefined           | Retained     |
|          |               | DATA_45    | Undefined   | Undefined           | Retained     |
|          |               | DATA_67    | Undefined   | Undefined           | Retained     |
|          |               | CONTROL1   | Initialized | Initialized         | Retained     |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[4].        | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |               | LAFM0      | Undefined   | Undefined           | Retained     |
|          |               | LAFM1      | Undefined   | Undefined           | Retained     |
|          |               | DATA_01    | Undefined   | Undefined           | Retained     |
|          |               | DATA_23    | Undefined   | Undefined           | Retained     |
|          |               | DATA_45    | Undefined   | Undefined           | Retained     |
|          |               | DATA_67    | Undefined   | Undefined           | Retained     |
|          |               | CONTROL1   | Initialized | Initialized         | Retained     |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[5].        | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |               | LAFM0      | Undefined   | Undefined           | Retained     |
|          |               | LAFM1      | Undefined   | Undefined           | Retained     |
|          |               | DATA_01    | Undefined   | Undefined           | Retained     |
|          |               | DATA_23    | Undefined   | Undefined           | Retained     |

|          |               |            | Reset State | Power               | r-Down State |
|----------|---------------|------------|-------------|---------------------|--------------|
| Module   | Register Name |            | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[5].        | DATA_45    | Undefined   | Undefined           | Retained     |
| (RCAN_A) |               | DATA_67    | Undefined   | Undefined           | Retained     |
|          |               | CONTROL1   | Initialized | Initialized         | Retained     |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[6].        | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |               | LAFM0      | Undefined   | Undefined           | Retained     |
|          |               | LAFM1      | Undefined   | Undefined           | Retained     |
|          |               | DATA_01    | Undefined   | Undefined           | Retained     |
|          |               | DATA_23    | Undefined   | Undefined           | Retained     |
|          |               | DATA_45    | Undefined   | Undefined           | Retained     |
|          |               | DATA_67    | Undefined   | Undefined           | Retained     |
|          |               | CONTROL1   | Initialized | Initialized         | Retained     |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[7].        | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |               | LAFM0      | Undefined   | Undefined           | Retained     |
|          |               | LAFM1      | Undefined   | Undefined           | Retained     |
|          |               | DATA_01    | Undefined   | Undefined           | Retained     |
|          |               | DATA_23    | Undefined   | Undefined           | Retained     |
|          |               | DATA_45    | Undefined   | Undefined           | Retained     |
|          |               | DATA_67    | Undefined   | Undefined           | Retained     |
|          |               | CONTROL1   | Initialized | Initialized         | Retained     |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[8].        | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |               | LAFM0      | Undefined   | Undefined           | Retained     |
|          |               | LAFM1      | Undefined   | Undefined           | Retained     |
|          |               | DATA_01    | Undefined   | Undefined           | Retained     |

|          |               |            | Reset State | Power-D             | own State |
|----------|---------------|------------|-------------|---------------------|-----------|
| Module   | Register Name |            | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[8].        | DATA_23    | Undefined   | Undefined           | Retained  |
| (RCAN_A) |               | DATA_45    | Undefined   | Undefined           | Retained  |
|          |               | DATA_67    | Undefined   | Undefined           | Retained  |
|          |               | CONTROL1   | Initialized | Initialized         | Retained  |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[9].        | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |               | LAFM0      | Undefined   | Undefined           | Retained  |
|          |               | LAFM1      | Undefined   | Undefined           | Retained  |
|          |               | DATA_01    | Undefined   | Undefined           | Retained  |
|          |               | DATA_23    | Undefined   | Undefined           | Retained  |
|          |               | DATA_45    | Undefined   | Undefined           | Retained  |
|          |               | DATA_67    | Undefined   | Undefined           | Retained  |
|          |               | CONTROL1   | Initialized | Initialized         | Retained  |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[10].       | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |               | LAFM0      | Undefined   | Undefined           | Retained  |
|          |               | LAFM1      | Undefined   | Undefined           | Retained  |
|          |               | DATA_01    | Undefined   | Undefined           | Retained  |
|          |               | DATA_23    | Undefined   | Undefined           | Retained  |
|          |               | DATA_45    | Undefined   | Undefined           | Retained  |
|          |               | DATA_67    | Undefined   | Undefined           | Retained  |
|          |               | CONTROL1   | Initialized | Initialized         | Retained  |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[11].       | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |               | LAFM0      | Undefined   | Undefined           | Retained  |
| -        |               | LAFM1      | Undefined   | Undefined           | Retained  |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[11]. | DATA_01    | Undefined   | Undefined           | Retained  |
| (RCAN_A) |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[12]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[13]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[14]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |

|          | Register Name |            | Reset State | Power-I             | Down State |
|----------|---------------|------------|-------------|---------------------|------------|
| Module   |               |            | Power-On    | Hardware<br>Standby | Sleep      |
| RCAN-TL1 | MB[14].       | LAFM1      | Undefined   | Undefined           | Retained   |
| (RCAN_A) |               | DATA_01    | Undefined   | Undefined           | Retained   |
|          |               | DATA_23    | Undefined   | Undefined           | Retained   |
|          |               | DATA_45    | Undefined   | Undefined           | Retained   |
|          |               | DATA_67    | Undefined   | Undefined           | Retained   |
|          |               | CONTROL1   | Initialized | Initialized         | Retained   |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained   |
|          | MB[15].       | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |               | LAFM0      | Undefined   | Undefined           | Retained   |
|          |               | LAFM1      | Undefined   | Undefined           | Retained   |
|          |               | DATA_01    | Undefined   | Undefined           | Retained   |
|          |               | DATA_23    | Undefined   | Undefined           | Retained   |
|          |               | DATA_45    | Undefined   | Undefined           | Retained   |
|          |               | DATA_67    | Undefined   | Undefined           | Retained   |
|          |               | CONTROL1   | Initialized | Initialized         | Retained   |
|          |               | TIMESTAMP  | Initialized | Initialized         | Retained   |
|          | MB[16].       | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |               | LAFM0      | Undefined   | Undefined           | Retained   |
|          |               | LAFM1      | Undefined   | Undefined           | Retained   |
|          |               | DATA_01    | Undefined   | Undefined           | Retained   |
|          |               | DATA_23    | Undefined   | Undefined           | Retained   |
|          |               | DATA_45    | Undefined   | Undefined           | Retained   |
|          |               | DATA_67    | Undefined   | Undefined           | Retained   |
|          |               | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[17].       | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |               | LAFM0      | Undefined   | Undefined           | Retained   |

|          |               |            | Reset State | Power-D             | own State |
|----------|---------------|------------|-------------|---------------------|-----------|
| Module   | Register Name |            | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[17].       | LAFM1      | Undefined   | Undefined           | Retained  |
| (RCAN_A) |               | DATA_01    | Undefined   | Undefined           | Retained  |
|          |               | DATA_23    | Undefined   | Undefined           | Retained  |
|          |               | DATA_45    | Undefined   | Undefined           | Retained  |
|          |               | DATA_67    | Undefined   | Undefined           | Retained  |
|          |               | CONTROL1   | Initialized | Initialized         | Retained  |
|          | MB[18].       | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |               | LAFM0      | Undefined   | Undefined           | Retained  |
|          |               | LAFM1      | Undefined   | Undefined           | Retained  |
|          |               | DATA_01    | Undefined   | Undefined           | Retained  |
|          |               | DATA_23    | Undefined   | Undefined           | Retained  |
|          |               | DATA_45    | Undefined   | Undefined           | Retained  |
|          |               | DATA_67    | Undefined   | Undefined           | Retained  |
|          |               | CONTROL1   | Initialized | Initialized         | Retained  |
|          | MB[19].       | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |               | LAFM0      | Undefined   | Undefined           | Retained  |
|          |               | LAFM1      | Undefined   | Undefined           | Retained  |
|          |               | DATA_01    | Undefined   | Undefined           | Retained  |
|          |               | DATA_23    | Undefined   | Undefined           | Retained  |
|          |               | DATA_45    | Undefined   | Undefined           | Retained  |
|          |               | DATA_67    | Undefined   | Undefined           | Retained  |
|          |               | CONTROL1   | Initialized | Initialized         | Retained  |
|          | MB[20].       | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |               | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |               | LAFM0      | Undefined   | Undefined           | Retained  |
|          |               | LAFM1      | Undefined   | Undefined           | Retained  |
|          |               | DATA_01    | Undefined   | Undefined           | Retained  |

|          |         |            | Reset State | Power               | r-Down State |
|----------|---------|------------|-------------|---------------------|--------------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[20]. | DATA_23    | Undefined   | Undefined           | Retained     |
| (RCAN_A) |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          | MB[21]. | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          | MB[22]. | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          | MB[23]. | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[23]. | DATA_67    | Undefined   | Undefined           | Retained  |
| (RCAN_A) |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          | MB[24]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[25]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[26]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[26]. | DATA_23    | Undefined   | Undefined           | Retained  |
| (RCAN_A) |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[27]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[28]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |

| Module   Register Name   Power-On   Standby   Sleep  |          |         |            | Reset State | Powe        | r-Down State |
|--|----------|---------|------------|-------------|-------------|--------------|
| (RCAN_A)  CONTROLO_L Undefined Undefined Retained LAFM0 Undefined Undefined Retained LAFM1 Undefined Undefined Retained DATA_01 Undefined Undefined Retained DATA_23 Undefined Undefined Retained DATA_45 Undefined Undefined Retained DATA_67 Undefined Undefined Retained CONTROL1 Initialized Initialized Retained TTT Initialized Initialized Retained TTCONTROL Initialized Initialized Retained CONTROLO_H Undefined Undefined Retained CONTROLO_L Undefined Undefined Retained LAFM0 Undefined Undefined Retained DATA_01 Undefined Undefined Retained DATA_01 Undefined Undefined Retained DATA_23 Undefined Undefined Retained DATA_23 Undefined Undefined Retained DATA_45 Undefined Undefined Retained DATA_45 Undefined Undefined Retained DATA_67 Undefined Undefined Retained DATA_67 Undefined Undefined Retained DATA_67 Undefined Undefined Retained TIMESTAMP Initialized Initialized Retained | Module   | Registe | r Name     | Power-On    |             | Sleep        |
| LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  TTT Initialized Initialized Retained  TTCONTROL Initialized Initialized Retained  MB[30]. CONTROLO_H Undefined Undefined Retained  CONTROLO_L Undefined Undefined Retained  LAFM0 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained                        | -        | MB[29]. | CONTROL0_H | Undefined   | Undefined   | Retained     |
| LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  TTT Initialized Initialized Retained  TTCONTROL Initialized Initialized Retained  TONTROL Initialized Undefined Retained  TONTROL Undefined Undefined Retained  CONTROLO_H Undefined Undefined Retained  LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Retained  TIMESTAMP Initialized Initialized Retained  | (RCAN_A) |         | CONTROL0_L | Undefined   | Undefined   | Retained     |
| DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TTT Initialized Initialized Retained  TTCONTROL Initialized Initialized Retained  MB[30]. CONTROL0_H Undefined Undefined Retained  CONTROL0_L Undefined Undefined Retained  LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained   |          |         | LAFM0      | Undefined   | Undefined   | Retained     |
| DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TTT Initialized Initialized Retained  TTCONTROL Initialized Initialized Retained  MB[30]. CONTROL0_H Undefined Undefined Retained  CONTROL0_L Undefined Undefined Retained  LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained   |          |         | LAFM1      | Undefined   | Undefined   | Retained     |
| DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TTT Initialized Initialized Retained  TTCONTROL Initialized Initialized Retained  MB[30]. CONTROLO_H Undefined Undefined Retained  CONTROLO_L Undefined Undefined Retained  LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained   |          |         | DATA_01    | Undefined   | Undefined   | Retained     |
| DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TTT Initialized Initialized Retained  TTCONTROL Initialized Initialized Retained  MB[30]. CONTROL0_H Undefined Undefined Retained  CONTROL0_L Undefined Undefined Retained  LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained   |          |         | DATA_23    | Undefined   | Undefined   | Retained     |
| CONTROL1 Initialized Initialized Retained TTT Initialized Initialized Retained TTCONTROL Initialized Initialized Retained  MB[30]. CONTROL0_H Undefined Undefined Retained CONTROL0_L Undefined Undefined Retained LAFM0 Undefined Undefined Retained LAFM1 Undefined Undefined Retained DATA_01 Undefined Undefined Retained DATA_23 Undefined Undefined Retained DATA_45 Undefined Undefined Retained DATA_67 Undefined Undefined Retained CONTROL1 Initialized Initialized Retained TIMESTAMP Initialized Initialized Retained  |          |         | DATA_45    | Undefined   | Undefined   | Retained     |
| TTT Initialized Initialized Retained TTCONTROL Initialized Initialized Retained  MB[30]. CONTROLO_H Undefined Undefined Retained  CONTROLO_L Undefined Undefined Retained  LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained   |          |         | DATA_67    | Undefined   | Undefined   | Retained     |
| TTCONTROL Initialized Initialized Retained  MB[30]. CONTROLO_H Undefined Undefined Retained  CONTROLO_L Undefined Undefined Retained  LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained  |          |         | CONTROL1   | Initialized | Initialized | Retained     |
| MB[30]. CONTROLO_H Undefined Undefined Retained  CONTROLO_L Undefined Undefined Retained  LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained  |          |         | TTT        | Initialized | Initialized | Retained     |
| CONTROLO_L Undefined Undefined Retained  LAFMO Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained   |          |         | TTCONTROL  | Initialized | Initialized | Retained     |
| LAFM0 Undefined Undefined Retained  LAFM1 Undefined Undefined Retained  DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained  |          | MB[30]. | CONTROL0_H | Undefined   | Undefined   | Retained     |
| LAFM1UndefinedUndefinedRetainedDATA_01UndefinedUndefinedRetainedDATA_23UndefinedUndefinedRetainedDATA_45UndefinedUndefinedRetainedDATA_67UndefinedUndefinedRetainedCONTROL1InitializedInitializedRetainedTIMESTAMPInitializedInitializedRetained   |          |         | CONTROL0_L | Undefined   | Undefined   | Retained     |
| DATA_01 Undefined Undefined Retained  DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained  |          |         | LAFM0      | Undefined   | Undefined   | Retained     |
| DATA_23 Undefined Undefined Retained  DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained  |          |         | LAFM1      | Undefined   | Undefined   | Retained     |
| DATA_45 Undefined Undefined Retained  DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained  |          |         | DATA_01    | Undefined   | Undefined   | Retained     |
| DATA_67 Undefined Undefined Retained  CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained  |          |         | DATA_23    | Undefined   | Undefined   | Retained     |
| CONTROL1 Initialized Initialized Retained  TIMESTAMP Initialized Initialized Retained  |          |         | DATA_45    | Undefined   | Undefined   | Retained     |
| TIMESTAMP Initialized Initialized Retained   |          |         | DATA_67    | Undefined   | Undefined   | Retained     |
|  |          |         | CONTROL1   | Initialized | Initialized | Retained     |
| TTT Initialized Initialized Retained   |          |         | TIMESTAMP  | Initialized | Initialized | Retained     |
| Industry   |          |         | TTT        | Initialized | Initialized | Retained     |
| MB[31]. CONTROL0_H Undefined Undefined Retained  |          | MB[31]. | CONTROL0_H | Undefined   | Undefined   | Retained     |
| CONTROL0_L Undefined Undefined Retained  |          |         | CONTROL0_L | Undefined   | Undefined   | Retained     |
| LAFM0 Undefined Undefined Retained   |          |         | LAFM0      | Undefined   | Undefined   | Retained     |
| LAFM1 Undefined Undefined Retained   |          |         | LAFM1      | Undefined   | Undefined   | Retained     |
| DATA_01 Undefined Undefined Retained   |          |         | DATA_01    | Undefined   | Undefined   | Retained     |
| DATA_23 Undefined Undefined Retained   |          |         | DATA_23    | Undefined   | Undefined   | Retained     |
| DATA_45 Undefined Undefined Retained   |          |         | DATA_45    | Undefined   | Undefined   | Retained     |
| DATA_67 Undefined Undefined Retained   |          |         | DATA_67    | Undefined   | Undefined   | Retained     |

|          |                  | Reset State | Powe                | r-Down State |
|----------|------------------|-------------|---------------------|--------------|
| Module   | Register Name    | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[31]. CONTROL1 | Initialized | Initialized         | Retained     |
| (RCAN_A) | TIMESTAMP        | Initialized | Initialized         | Retained     |
|          | MBESR            | Initialized | Initialized         | Retained     |
|          | MBECR            | Initialized | Initialized         | Retained     |
| RCAN-TL1 | MCR              | Initialized | Initialized         | Retained     |
| (RCAN_B) | GSR              | Initialized | Initialized         | Retained     |
|          | BCR1             | Initialized | Initialized         | Retained     |
|          | BCR0             | Initialized | Initialized         | Retained     |
|          | IRR              | Initialized | Initialized         | Retained     |
|          | IMR              | Initialized | Initialized         | Retained     |
|          | TEC/REC          | Initialized | Initialized         | Retained     |
|          | TXPR1            | Initialized | Initialized         | Retained     |
|          | TXPR0            | Initialized | Initialized         | Retained     |
|          | TXCR1            | Initialized | Initialized         | Retained     |
|          | TXCR0            | Initialized | Initialized         | Retained     |
|          | TXACK1           | Initialized | Initialized         | Retained     |
|          | TXACK0           | Initialized | Initialized         | Retained     |
|          | ABACK1           | Initialized | Initialized         | Retained     |
|          | ABACK0           | Initialized | Initialized         | Retained     |
|          | RXPR1            | Initialized | Initialized         | Retained     |
|          | RXPR0            | Initialized | Initialized         | Retained     |
|          | RFPR1            | Initialized | Initialized         | Retained     |
|          | RFPR0            | Initialized | Initialized         | Retained     |
|          | MBIMR1           | Initialized | Initialized         | Retained     |
|          | MBIMR0           | Initialized | Initialized         | Retained     |
|          | UMSR1            | Initialized | Initialized         | Retained     |
|          | UMSR0            | Initialized | Initialized         | Retained     |
|          | TTCR0            | Initialized | Initialized         | Retained     |
|          | CMAX_TEW         | Initialized | Initialized         | Retained     |
|          | RFTROFF          | Initialized | Initialized         | Retained     |

|          |         |            | Reset State | Power-I             | Down State |
|----------|---------|------------|-------------|---------------------|------------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep      |
| RCAN-TL1 | TSR     |            | Initialized | Initialized         | Retained   |
| (RCAN_B) | CCR     |            | Initialized | Initialized         | Retained   |
|          | TCNTR   |            | Initialized | Initialized         | Retained   |
|          | CYCTR   |            | Initialized | Initialized         | Retained   |
|          | RFMK    |            | Initialized | Initialized         | Retained   |
|          | TCMR0   |            | Initialized | Initialized         | Retained   |
|          | TCMR1   |            | Initialized | Initialized         | Retained   |
|          | TCMR2   |            | Initialized | Initialized         | Retained   |
|          | TTTSEL  | _          | Initialized | Initialized         | Retained   |
|          | MB[0].  | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained   |
|          | MB[1].  | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained   |

|          |         |            | Reset State | Powe                | r-Down State |
|----------|---------|------------|-------------|---------------------|--------------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[2].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
| (RCAN_B) |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[3].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[4].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |

|          |         |            | Reset State | Power               | -Down State |
|----------|---------|------------|-------------|---------------------|-------------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep       |
| RCAN-TL1 | MB[5].  | CONTROL0_H | Undefined   | Undefined           | Retained    |
| (RCAN_B) |         | CONTROL0_L | Undefined   | Undefined           | Retained    |
|          |         | LAFM0      | Undefined   | Undefined           | Retained    |
|          |         | LAFM1      | Undefined   | Undefined           | Retained    |
|          |         | DATA_01    | Undefined   | Undefined           | Retained    |
|          |         | DATA_23    | Undefined   | Undefined           | Retained    |
|          |         | DATA_45    | Undefined   | Undefined           | Retained    |
|          |         | DATA_67    | Undefined   | Undefined           | Retained    |
|          |         | CONTROL1   | Initialized | Initialized         | Retained    |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained    |
|          | MB[6].  | CONTROL0_H | Undefined   | Undefined           | Retained    |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained    |
|          |         | LAFM0      | Undefined   | Undefined           | Retained    |
|          |         | LAFM1      | Undefined   | Undefined           | Retained    |
|          |         | DATA_01    | Undefined   | Undefined           | Retained    |
|          |         | DATA_23    | Undefined   | Undefined           | Retained    |
|          |         | DATA_45    | Undefined   | Undefined           | Retained    |
|          |         | DATA_67    | Undefined   | Undefined           | Retained    |
|          |         | CONTROL1   | Initialized | Initialized         | Retained    |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained    |
|          | MB[7].  | CONTROL0_H | Undefined   | Undefined           | Retained    |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained    |
|          |         | LAFM0      | Undefined   | Undefined           | Retained    |
|          |         | LAFM1      | Undefined   | Undefined           | Retained    |
|          |         | DATA_01    | Undefined   | Undefined           | Retained    |
|          |         | DATA_23    | Undefined   | Undefined           | Retained    |
|          |         | DATA_45    | Undefined   | Undefined           | Retained    |
|          |         | DATA_67    | Undefined   | Undefined           | Retained    |
|          |         | CONTROL1   | Initialized | Initialized         | Retained    |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained    |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[8].  | CONTROL0_H | Undefined   | Undefined           | Retained  |
| (RCAN_B) |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[9].  | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[10]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |

|          |         |            | Reset State | Power-              | Down State |
|----------|---------|------------|-------------|---------------------|------------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep      |
| RCAN-TL1 | MB[11]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
| (RCAN_B) |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained   |
|          | MB[12]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained   |
|          | MB[13]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained   |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[14]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
| (RCAN_B) |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[15]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[16]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |

|          |         |            | Reset State | Power-I             | Down State |
|----------|---------|------------|-------------|---------------------|------------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep      |
| RCAN-TL1 | MB[17]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
| (RCAN_B) |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[18]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[19]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[20]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[20]. | LAFM0      | Undefined   | Undefined           | Retained  |
| (RCAN_B) |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          | MB[21]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          | MB[22]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          | MB[23]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[23]. | DATA_01    | Undefined   | Undefined           | Retained  |
| (RCAN_B) |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          | MB[24]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[25]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[26]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[26]. | LAFM0      | Undefined   | Undefined           | Retained  |
| (RCAN_B) |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[27]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[28]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |

|          |         |            | Reset State | Power               | r-Down State |
|----------|---------|------------|-------------|---------------------|--------------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[28]. | TTT        | Initialized | Initialized         | Retained     |
| (RCAN_B) |         | TTCONTROL  | Initialized | Initialized         | Retained     |
|          | MB[29]. | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TTT        | Initialized | Initialized         | Retained     |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained     |
|          | MB[30]. | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          |         | TTT        | Initialized | Initialized         | Retained     |
|          | MB[31]. | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |

|          |         |           | Reset State | Powe                | r-Down State |
|----------|---------|-----------|-------------|---------------------|--------------|
| Module   | Registe | er Name   | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[31]. | DATA_45   | Undefined   | Undefined           | Retained     |
| (RCAN_B) |         | DATA_67   | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1  | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP | Initialized | Initialized         | Retained     |
|          | MBESR   | l         | Initialized | Initialized         | Retained     |
|          | MBECR   | }         | Initialized | Initialized         | Retained     |
| RCAN-TL1 | MCR     |           | Initialized | Initialized         | Retained     |
| (RCAN_C) | GSR     |           | Initialized | Initialized         | Retained     |
|          | BCR1    |           | Initialized | Initialized         | Retained     |
|          | BCR0    |           | Initialized | Initialized         | Retained     |
|          | IRR     |           | Initialized | Initialized         | Retained     |
|          | IMR     |           | Initialized | Initialized         | Retained     |
|          | TEC/RE  | C         | Initialized | Initialized         | Retained     |
|          | TXPR1   |           | Initialized | Initialized         | Retained     |
|          | TXPR0   |           | Initialized | Initialized         | Retained     |
|          | TXCR1   |           | Initialized | Initialized         | Retained     |
|          | TXCR0   |           | Initialized | Initialized         | Retained     |
|          | TXACK   | 1         | Initialized | Initialized         | Retained     |
|          | TXACK   | 0         | Initialized | Initialized         | Retained     |
|          | ABACK   | 1         | Initialized | Initialized         | Retained     |
|          | ABACK   | 0         | Initialized | Initialized         | Retained     |
|          | RXPR1   |           | Initialized | Initialized         | Retained     |
|          | RXPR0   |           | Initialized | Initialized         | Retained     |
|          | RFPR1   |           | Initialized | Initialized         | Retained     |
|          | RFPR0   |           | Initialized | Initialized         | Retained     |
|          | MBIMR   | 1         | Initialized | Initialized         | Retained     |
|          | MBIMR   |           | Initialized | Initialized         | Retained     |
|          | UMSR1   |           | Initialized | Initialized         | Retained     |
|          | UMSR0   | <u> </u>  | Initialized | Initialized         | Retained     |
|          | TTCR0   |           | Initialized | Initialized         | Retained     |
|          | CMAX_   | IEW       | Initialized | Initialized         | Retained     |

|          |         |            | Reset State | Power               | r-Down State |
|----------|---------|------------|-------------|---------------------|--------------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | RFTRO   | FF         | Initialized | Initialized         | Retained     |
| (RCAN_C) | TSR     |            | Initialized | Initialized         | Retained     |
|          | CCR     |            | Initialized | Initialized         | Retained     |
|          | TCNTR   |            | Initialized | Initialized         | Retained     |
|          | CYCTR   | 1          | Initialized | Initialized         | Retained     |
|          | RFMK    |            | Initialized | Initialized         | Retained     |
|          | TCMR0   |            | Initialized | Initialized         | Retained     |
|          | TCMR1   |            | Initialized | Initialized         | Retained     |
|          | TCMR2   |            | Initialized | Initialized         | Retained     |
|          | TTTSEI  | <u> </u>   | Initialized | Initialized         | Retained     |
|          | MB[0].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[1].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |

|          |         |            | Reset State | Powe                | r-Down State |
|----------|---------|------------|-------------|---------------------|--------------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[2].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
| (RCAN_C) |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[3].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[4].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |

|          |         |            | Reset State | Powe                | r-Down State |
|----------|---------|------------|-------------|---------------------|--------------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[4].  | CONTROL1   | Initialized | Initialized         | Retained     |
| (RCAN_C) |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[5].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[6].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[7].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |

|          |         |            | Reset State | Power               | r-Down State |
|----------|---------|------------|-------------|---------------------|--------------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[7].  | DATA_67    | Undefined   | Undefined           | Retained     |
| (RCAN_C) |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[8].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[9].  | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MB[10]. | CONTROL0_H | Undefined   | Undefined           | Retained     |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[10]. | DATA_45    | Undefined   | Undefined           | Retained  |
| (RCAN_C) |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[11]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[12]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[13]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[13]. | DATA_23    | Undefined   | Undefined           | Retained  |
| (RCAN_C) |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[14]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[15]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          | MB[16]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |

|          |         |            | Reset State | Power-l             | Down State |
|----------|---------|------------|-------------|---------------------|------------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep      |
| RCAN-TL1 | MB[16]. | DATA_01    | Undefined   | Undefined           | Retained   |
| (RCAN_C) |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[17]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[18]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[19]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |

|          |         |            | Reset State | Power-              | Down State |
|----------|---------|------------|-------------|---------------------|------------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep      |
| RCAN-TL1 | MB[19]. | DATA_45    | Undefined   | Undefined           | Retained   |
| (RCAN_C) |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[20]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[21]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |
|          | MB[22]. | CONTROL0_H | Undefined   | Undefined           | Retained   |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained   |
|          |         | LAFM0      | Undefined   | Undefined           | Retained   |
|          |         | LAFM1      | Undefined   | Undefined           | Retained   |
|          |         | DATA_01    | Undefined   | Undefined           | Retained   |
|          |         | DATA_23    | Undefined   | Undefined           | Retained   |
|          |         | DATA_45    | Undefined   | Undefined           | Retained   |
|          |         | DATA_67    | Undefined   | Undefined           | Retained   |
|          |         | CONTROL1   | Initialized | Initialized         | Retained   |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[23]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
| (RCAN_C) |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          | MB[24]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[25]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[25]. | TTT        | Initialized | Initialized         | Retained  |
| (RCAN_C) |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[26]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[27]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[28]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |

|          |         |            | Reset State | Power-D             | own State |
|----------|---------|------------|-------------|---------------------|-----------|
| Module   | Registe | r Name     | Power-On    | Hardware<br>Standby | Sleep     |
| RCAN-TL1 | MB[28]. | DATA_23    | Undefined   | Undefined           | Retained  |
| (RCAN_C) |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[29]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |
|          |         | TTCONTROL  | Initialized | Initialized         | Retained  |
|          | MB[30]. | CONTROL0_H | Undefined   | Undefined           | Retained  |
|          |         | CONTROL0_L | Undefined   | Undefined           | Retained  |
|          |         | LAFM0      | Undefined   | Undefined           | Retained  |
|          |         | LAFM1      | Undefined   | Undefined           | Retained  |
|          |         | DATA_01    | Undefined   | Undefined           | Retained  |
|          |         | DATA_23    | Undefined   | Undefined           | Retained  |
|          |         | DATA_45    | Undefined   | Undefined           | Retained  |
|          |         | DATA_67    | Undefined   | Undefined           | Retained  |
|          |         | CONTROL1   | Initialized | Initialized         | Retained  |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained  |
|          |         | TTT        | Initialized | Initialized         | Retained  |

|          |         |            | Reset State | Powe                | r-Down State |
|----------|---------|------------|-------------|---------------------|--------------|
| Module   | Registe | er Name    | Power-On    | Hardware<br>Standby | Sleep        |
| RCAN-TL1 | MB[31]. | CONTROL0_H | Undefined   | Undefined           | Retained     |
| (RCAN_C) |         | CONTROL0_L | Undefined   | Undefined           | Retained     |
|          |         | LAFM0      | Undefined   | Undefined           | Retained     |
|          |         | LAFM1      | Undefined   | Undefined           | Retained     |
|          |         | DATA_01    | Undefined   | Undefined           | Retained     |
|          |         | DATA_23    | Undefined   | Undefined           | Retained     |
|          |         | DATA_45    | Undefined   | Undefined           | Retained     |
|          |         | DATA_67    | Undefined   | Undefined           | Retained     |
|          |         | CONTROL1   | Initialized | Initialized         | Retained     |
|          |         | TIMESTAMP  | Initialized | Initialized         | Retained     |
|          | MBESR   | 1          | Initialized | Initialized         | Retained     |
|          | MBECF   | ?          | Initialized | Initialized         | Retained     |
| ADC      | ADRD0   |            | Initialized | Initialized         | Retained     |
| (ADC_A)  | ADR0    |            | Initialized | Initialized         | Retained     |
|          | ADR1    |            | Initialized | Initialized         | Retained     |
|          | ADR2    |            | Initialized | Initialized         | Retained     |
|          | ADR3    |            | Initialized | Initialized         | Retained     |
|          | ADR4    |            | Initialized | Initialized         | Retained     |
|          | ADR5    |            | Initialized | Initialized         | Retained     |
|          | ADR6    |            | Initialized | Initialized         | Retained     |
|          | ADR7    |            | Initialized | Initialized         | Retained     |
|          | ADR8    |            | Initialized | Initialized         | Retained     |
|          | ADR9    |            | Initialized | Initialized         | Retained     |
|          | ADR10   |            | Initialized | Initialized         | Retained     |
|          | ADR11   |            | Initialized | Initialized         | Retained     |
|          | ADR12   |            | Initialized | Initialized         | Retained     |
|          | ADR13   |            | Initialized | Initialized         | Retained     |
|          | ADR14   |            | Initialized | Initialized         | Retained     |
|          | ADR15   |            | Initialized | Initialized         | Retained     |
|          | ADR16   |            | Initialized | Initialized         | Retained     |
|          | ADR17   |            | Initialized | Initialized         | Retained     |

|                |               | Reset State | Powe                | r-Down State |
|----------------|---------------|-------------|---------------------|--------------|
| Module         | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ADC            | ADR18         | Initialized | Initialized         | Retained     |
| (ADC_A)        | ADR19         | Initialized | Initialized         | Retained     |
|                | ADR20         | Initialized | Initialized         | Retained     |
|                | ADR21         | Initialized | Initialized         | Retained     |
|                | ADR22         | Initialized | Initialized         | Retained     |
|                | ADR23         | Initialized | Initialized         | Retained     |
|                | ADR24         | Initialized | Initialized         | Retained     |
|                | ADR25         | Initialized | Initialized         | Retained     |
|                | ADR26         | Initialized | Initialized         | Retained     |
|                | ADR27         | Initialized | Initialized         | Retained     |
| ADC            | ADRD1         | Initialized | Initialized         | Retained     |
| (ADC_B)        | ADR40         | Initialized | Initialized         | Retained     |
|                | ADR41         | Initialized | Initialized         | Retained     |
|                | ADR42         | Initialized | Initialized         | Retained     |
|                | ADR43         | Initialized | Initialized         | Retained     |
|                | ADR44         | Initialized | Initialized         | Retained     |
|                | ADR45         | Initialized | Initialized         | Retained     |
|                | ADR46         | Initialized | Initialized         | Retained     |
|                | ADR47         | Initialized | Initialized         | Retained     |
|                | ADR48         | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_A) | ADCSR0        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADCSR1        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_A) | ADREF0        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADREF1        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_A) | ADTRE0        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADTRE1        | Initialized | Initialized         | Retained     |

|                |               | Reset State | Powe                | r-Down State |
|----------------|---------------|-------------|---------------------|--------------|
| Module         | Register Name | Power-On    | Hardware<br>Standby | Sleep        |
| ADC<br>(ADC_A) | ADTRF0        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADTRF1        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_A) | ADTRS0        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADTRS1        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_A) | ADSTRG0       | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADSTRG1       | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_A) | ADTRD0        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADTRD1        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_A) | ADADS0        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADADS1        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_A) | ADADC0        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADADC1        | Initialized | Initialized         | Retained     |
| ADC (ABC A)    | ADANS0        | Initialized | Initialized         | Retained     |
| (ADC_A)        | ADANS1        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADANS3        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_A) | ADCER0        | Initialized | Initialized         | Retained     |
| ADC<br>(ADC_B) | ADCER1        | Initialized | Initialized         | Retained     |
| JTAG           | SDIR          | Undefined   | Undefined           | Retained     |
|                | SDID          | Retained    | Retained            | Retained     |
|                | SDBPR         | Undefined   | Undefined           | Retained     |

|        |               | Reset State | Power-Down State    |          |
|--------|---------------|-------------|---------------------|----------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep    |
| JTAG   | SDBSR         | Undefined   | Undefined           | Retained |
| AUD-II | AUCSR         | Initialized | Initialized         | Retained |
|        | AUWASR        | Undefined   | Undefined           | Retained |
|        | AUWAER        | Undefined   | Undefined           | Retained |
|        | AUWBSR        | Undefined   | Undefined           | Retained |
|        | AUWBER        | Undefined   | Undefined           | Retained |
|        | AUECSR        | Initialized | Initialized         | Retained |
| PFC    | PAIOR         | Initialized | Initialized         | Retained |
|        | PACR4         | Initialized | Initialized         | Retained |
|        | PACR3         | Initialized | Initialized         | Retained |
|        | PACR2         | Initialized | Initialized         | Retained |
|        | PACR1         | Initialized | Initialized         | Retained |
|        | PBIOR         | Initialized | Initialized         | Retained |
|        | PBCR4         | Initialized | Initialized         | Retained |
|        | PBCR3         | Initialized | Initialized         | Retained |
|        | PBCR2         | Initialized | Initialized         | Retained |
|        | PBCR1         | Initialized | Initialized         | Retained |
|        | PCIOR         | Initialized | Initialized         | Retained |
|        | PCCR4         | Initialized | Initialized         | Retained |
|        | PCCR3         | Initialized | Initialized         | Retained |
|        | PCCR2         | Initialized | Initialized         | Retained |
|        | PCCR1         | Initialized | Initialized         | Retained |
|        | PDIOR         | Initialized | Initialized         | Retained |
|        | PDCR2         | Initialized | Initialized         | Retained |
|        | PDCR1         | Initialized | Initialized         | Retained |
|        | PEIOR         | Initialized | Initialized         | Retained |
|        | PECR2         | Initialized | Initialized         | Retained |
|        | PECR1         | Initialized | Initialized         | Retained |
|        | PFIOR         | Initialized | Initialized         | Retained |
|        | PFCR2         | Initialized | Initialized         | Retained |
|        | PFCR1         | Initialized | Initialized         | Retained |

|          |               | Reset State | Power-Down State    |          |  |
|----------|---------------|-------------|---------------------|----------|--|
| Module   | Register Name | Power-On    | Hardware<br>Standby | Sleep    |  |
| PFC      | PGIOR         | Initialized | Initialized         | Retained |  |
|          | PGCR2         | Initialized | Initialized         | Retained |  |
|          | PGCR1         | Initialized | Initialized         | Retained |  |
|          | PHIOR         | Initialized | Initialized         | Retained |  |
|          | PHCR          | Initialized | Initialized         | Retained |  |
|          | PJIOR         | Initialized | Initialized         | Retained |  |
|          | PJCR2         | Initialized | Initialized         | Retained |  |
|          | PJCR1         | Initialized | Initialized         | Retained |  |
|          | PKIOR         | Initialized | Initialized         | Retained |  |
|          | PKCR2         | Initialized | Initialized         | Retained |  |
|          | PKCR1         | Initialized | Initialized         | Retained |  |
|          | PLIOR         | Initialized | Initialized         | Retained |  |
|          | PLCR2         | Initialized | Initialized         | Retained |  |
|          | PLCR1         | Initialized | Initialized         | Retained |  |
| I/O port | PADR          | Initialized | Initialized         | Retained |  |
|          | PAPR          | Pin value   | Pin value           | Retained |  |
|          | PBDR          | Initialized | Initialized         | Retained |  |
|          | PBPR          | Pin value   | Pin value           | Retained |  |
|          | PBIR          | Initialized | Initialized         | Retained |  |
|          | PBDSR         | Initialized | Initialized         | Retained |  |
|          | PBPSR         | Initialized | Initialized         | Retained |  |
|          | PCDR          | Initialized | Initialized         | Retained |  |
|          | PCPR          | Pin value   | Pin value           | Retained |  |
|          | PDDR          | Initialized | Initialized         | Retained |  |
|          | PDPR          | Pin value   | Pin value           | Retained |  |
|          | PDIR          | Initialized | Initialized         | Retained |  |
|          | PEDR          | Initialized | Initialized         | Retained |  |
|          | PEPR          | Pin value   | Pin value           | Retained |  |
|          | PEIR          | Initialized | Initialized         | Retained |  |
|          | PEDSR         | Initialized | Initialized         | Retained |  |
|          | PFDR          | Initialized | Initialized         | Retained |  |

| Module   Register Name   Power-On   Standby   Sleep  |          |               | Reset State | Power-Down  | State    |
|--|----------|---------------|-------------|-------------|----------|
| PFIR Initialized Initialized Retained PFDSR Initialized Initialized Retained PFPSR Initialized Initialized Retained PGDR Initialized Initialized Retained PGDR Initialized Initialized Retained PGDR Pin value Pin value Retained PGDR Initialized Initialized Retained PGDSR Initialized Initialized Retained PGDSR Initialized Initialized Retained PGER Initialized Initialized Retained PHDR Initialized Initialized Retained PHDR Pin value Pin value Retained PHDR Initialized Initialized Retained PJDR Initialized Initialized Retained PJDR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJDSR Initialized Initialized Retained PHOR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained   | Module   | Register Name | Power-On    |             | Sleep    |
| PFDSR Initialized Initialized Retained PFPSR Initialized Initialized Retained PGDR Initialized Initialized Retained PGDR Pin value Pin value Retained PGPR Pin value Pin value Retained PGRR Initialized Initialized Retained PGDSR Initialized Initialized Retained PGER Initialized Initialized Retained PHDR Initialized Initialized Retained PHPR Pin value Pin value Retained PJDR Initialized Initialized Retained PJPR Pin value Pin value Retained PJPR Pin value Pin value Retained PJRR Initialized Initialized Retained PJPSR Initialized Initialized Retained PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKPR Pin value Pin value Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR PLIR Initialized Initialized Retained PLIR Initialized Initialized Retained MISG MISRCDR Undefined Undefined Retained   | I/O port | PFPR          | Pin value   | Pin value   | Retained |
| PFPSR Initialized Initialized Retained PGDR Pin value Pin value Retained PGPR Pin value Pin value Retained PGIR Initialized Initialized Retained PGIR Initialized Initialized Retained PGDSR Initialized Initialized Retained PGER Initialized Initialized Retained PHDR Initialized Initialized Retained PHPR Pin value Pin value Retained PJDR Initialized Initialized Retained PJDR Initialized Initialized Retained PJPR Pin value Pin value Retained PJIR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJDSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Pin value Pin value Retained PLDR Pin value Pin value Retained PLDR Initialized Initialized Retained PLIR Initialized Initialized Retained MISGODR Undefined Undefined Retained MISGODR Undefined Undefined Retained   |          | PFIR          | Initialized | Initialized | Retained |
| PGDR Initialized Initialized Retained PGPR Pin value Pin value Retained PGIR Initialized Initialized Retained PGDSR Initialized Initialized Retained PGER Initialized Initialized Retained PHDR Initialized Initialized Retained PHDR Pin value Pin value Retained PHPR Pin value Pin value Retained PJDR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJDSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKDR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLIR Initialized Initialized Retained MISACDR Undefined Undefined Retained MISACDR Initialized Initialized Retained   |          | PFDSR         | Initialized | Initialized | Retained |
| PGPR Pin value Pin value Retained PGIR Initialized Initialized Retained PGDSR Initialized Initialized Retained PGER Initialized Initialized Retained PHDR Initialized Initialized Retained PHPR Pin value Pin value Retained PJDR Initialized Initialized Retained PJPR Pin value Pin value Retained PJPR Pin value Pin value Retained PJPR Pin value Pin value Retained PJPR Initialized Initialized Retained PJPSR Initialized Initialized Retained PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLIR Initialized Initialized Retained MISGCR Initialized Initialized Retained Retained   |          | PFPSR         | Initialized | Initialized | Retained |
| PGIR Initialized Initialized Retained PGDSR Initialized Initialized Retained PGER Initialized Initialized Retained PHDR Initialized Initialized Retained PHDR Pin value Pin value Retained PHPR Pin value Pin value Retained PJDR Initialized Initialized Retained PJDR Initialized Initialized Retained PJPR Pin value Pin value Retained PJIR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained MISG MISRCDR Undefined Undefined Retained MISRCR Initialized Initialized Retained   |          | PGDR          | Initialized | Initialized | Retained |
| PGDSR Initialized Initialized Retained PGER Initialized Initialized Retained PHDR Initialized Initialized Retained PHPR Pin value Pin value Retained PJDR Initialized Initialized Retained PJDR Initialized Initialized Retained PJPR Pin value Pin value Retained PJPR Pin value Pin value Retained PJIR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLIR Initialized Initialized Retained MISG MISRCDR Undefined Undefined Retained MISG Retained Initialized Retained Retained  |          | PGPR          | Pin value   | Pin value   | Retained |
| PGER Initialized Initialized Retained PHDR Initialized Initialized Retained PHPR Pin value Pin value Retained PJDR Initialized Initialized Retained PJDR Pin value Pin value Retained PJPR Pin value Pin value Retained PJIR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJDSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKDR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained  MISG MISRCDR Undefined Undefined Retained  MISG Initialized Initialized Retained  |          | PGIR          | Initialized | Initialized | Retained |
| PHDR Initialized Initialized Retained PHPR Pin value Pin value Retained PJDR Initialized Initialized Retained PJPR Pin value Pin value Retained PJPR Pin value Pin value Retained PJIR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained MISRCR Initialized Initialized Retained Retained  |          | PGDSR         | Initialized | Initialized | Retained |
| PHPR Pin value Pin value Retained PJDR Initialized Initialized Retained PJPR Pin value Pin value Retained PJIR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJDSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLIR Initialized Initialized Retained MISRCR Initialized Initialized Retained Retained  |          | PGER          | Initialized | Initialized | Retained |
| PJDR Initialized Initialized Retained PJPR Pin value Pin value Retained PJIR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained PLIR Initialized Initialized Retained MISG MISRCDR Undefined Undefined Retained Initialized Retained  |          | PHDR          | Initialized | Initialized | Retained |
| PJPR Pin value Pin value Retained PJIR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained  MISG MISRCDR Undefined Undefined Retained MISRCR Initialized Initialized Retained   |          | PHPR          | Pin value   | Pin value   | Retained |
| PJIR Initialized Initialized Retained PJDSR Initialized Initialized Retained PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained MISRCR Initialized Initialized Retained Retained Retained   |          | PJDR          | Initialized | Initialized | Retained |
| PJDSR Initialized Initialized Retained PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained MISG MISRCDR Undefined Undefined Retained MISRCR Initialized Initialized Retained   |          | PJPR          | Pin value   | Pin value   | Retained |
| PJPSR Initialized Initialized Retained PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained MISG MISRCDR Undefined Initialized Retained MISRCR Initialized Initialized Retained Retained   |          | PJIR          | Initialized | Initialized | Retained |
| PKDR Initialized Initialized Retained PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained RETAINED  MISCR Initialized Initialized Retained Initialized Retained Retained Retained INITIALIZED INIT |          | PJDSR         | Initialized | Initialized | Retained |
| PKPR Pin value Pin value Retained PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained RETAINED RETAINED MISG MISCOR Undefined Undefined Retained MISRCR Initialized Initialized Retained MISRCR Initialized Initialized Retained MISRCR Initialized Initialized Retained   |          | PJPSR         | Initialized | Initialized | Retained |
| PKIR Initialized Initialized Retained PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained CKCR Initialized Initialized Retained MISG MISRCDR Undefined Undefined Retained MISRCR Initialized Initialized Retained  |          | PKDR          | Initialized | Initialized | Retained |
| PKDSR Initialized Initialized Retained PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained CKCR Initialized Initialized Retained MISRCDR Undefined Undefined Retained MISRCR Initialized Initialized Retained   |          | PKPR          | Pin value   | Pin value   | Retained |
| PKPSR Initialized Initialized Retained PLDR Initialized Initialized Retained PLPR Pin value Pin value Retained PLIR Initialized Initialized Retained CKCR Initialized Initialized Retained MISG MISRCDR Undefined Undefined Retained MISRCR Initialized Initialized Retained   |          | PKIR          | Initialized | Initialized | Retained |
| PLDRInitializedInitializedRetainedPLPRPin valuePin valueRetainedPLIRInitializedInitializedRetainedCKCRInitializedInitializedRetainedMISGMISRCDRUndefinedUndefinedRetainedMISRCRInitializedInitializedRetained  |          | PKDSR         | Initialized | Initialized | Retained |
| PLPR     Pin value     Pin value     Retained       PLIR     Initialized     Initialized     Retained       CKCR     Initialized     Initialized     Retained       MISRCDR     Undefined     Undefined     Retained       MISRCR     Initialized     Initialized     Retained   |          | PKPSR         | Initialized | Initialized | Retained |
| PLIR Initialized Initialized Retained  CKCR Initialized Initialized Retained  MISG MISRCDR Undefined Undefined Retained  MISRCR Initialized Initialized Retained   |          | PLDR          | Initialized | Initialized | Retained |
| CKCR Initialized Initialized Retained  MISG MISRCDR Undefined Undefined Retained  MISRCR Initialized Initialized Retained  |          | PLPR          | Pin value   | Pin value   | Retained |
| MISRCDR Undefined Undefined Retained  MISRCR Initialized Initialized Retained  |          | PLIR          | Initialized | Initialized | Retained |
| MISRCR Initialized Initialized Retained  |          | CKCR          | Initialized | Initialized | Retained |
|  | MISG     | MISRCDR       | Undefined   | Undefined   | Retained |
| MICD Initialized Initialized Detained  |          | MISRCR        | Initialized | Initialized | Retained |
| MIGH IIIItalizeu IIIItalizeu Hetairieu   |          | MISR          | Initialized | Initialized | Retained |

|        |               | Reset State | Power-Down          | State    |
|--------|---------------|-------------|---------------------|----------|
| Module | Register Name | Power-On    | Hardware<br>Standby | Sleep    |
| ROM/   | FPMON         | Initialized | Initialized         | Retained |
| EEPROM | FMODR         | Initialized | Initialized         | Retained |
|        | FASTAT        | Initialized | Initialized         | Retained |
|        | FAEINT        | Initialized | Initialized         | Retained |
|        | ROMMAT        | Initialized | Initialized         | Retained |
|        | FCURAME       | Initialized | Initialized         | Retained |
|        | FSTATR0       | Initialized | Initialized         | Retained |
|        | FSTATR1       | Initialized | Initialized         | Retained |
|        | FENTRYR       | Initialized | Initialized         | Retained |
|        | FPROTR        | Initialized | Initialized         | Retained |
|        | FRESETR       | Initialized | Initialized         | Retained |
|        | FCMDR         | Initialized | Initialized         | Retained |
|        | FRAMECCR      | Initialized | Initialized         | Retained |
|        | FCPSR         | Initialized | Initialized         | Retained |
|        | EEPBCCNT      | Initialized | Initialized         | Retained |
|        | FPESTAT       | Initialized | Initialized         | Retained |
|        | EEPBCSTAT     | Initialized | Initialized         | Retained |
|        | EEPRE0        | Initialized | Initialized         | Retained |
|        | EEPRE1        | Initialized | Initialized         | Retained |
|        | EEPWE0        | Initialized | Initialized         | Retained |
|        | EEPWE1        | Initialized | Initialized         | Retained |
|        | EEPMAT        | Initialized | Initialized         | Retained |
| ROMC   | RCCR          | Initialized | Initialized         | Retained |
|        | RCCR2         | Initialized | Initialized         | Retained |
| RAM    | RAMEN0        | Initialized | Initialized         | Retained |
|        | RAMWEN0       | Initialized | Initialized         | Retained |
|        | RAMECC        | Initialized | Initialized         | Retained |
|        | RAMERR        | Initialized | Initialized         | Retained |
|        | RAMINT        | Initialized | Initialized         | Retained |
|        | RAMACYC       | Initialized | Initialized         | Retained |
|        | RAMEN1        | Initialized | Initialized         | Retained |
|        |               |             |                     |          |

|                 |               | Reset State | Power-Down State    |          |
|-----------------|---------------|-------------|---------------------|----------|
| Module          | Register Name | Power-On    | Hardware<br>Standby | Sleep    |
| RAM             | RAMWEN1       | Initialized | Initialized         | Retained |
| Power-down mode | STBCR         | Initialized | Initialized         | Retained |

# Section 32 Electrical Characteristics

This section shows the electrical characteristics of the R5F72546R.

## 32.1 Absolute Maximum Ratings

Table 32.1 shows the absolute maximum ratings.

**Table 32.1 Absolute Maximum Ratings** 

| Item                            |  |  | Symbol           | Rating                          | Unit | Remarks                                  |
|---------------------------------|--|--|------------------|---------------------------------|------|--|
| Power                           | V <sub>cc</sub> and  | PLLV <sub>cc</sub> pins  | V <sub>cc</sub>  | -0.3 to +4.3                    | V    |  |
| supply<br>voltage* <sup>1</sup> | PV <sub>cc</sub> 1 a<br>PV <sub>cc</sub> 2 p                               |  | PV <sub>cc</sub> | -0.3 to +6.5                    | V    |  |
| Input<br>voltage                | V <sub>cc</sub><br>power   | EXTAL, JTAG,<br>AUD-II pins                                      | Vin              | $-0.3$ to $V_{cc} + 0.3$        | V    | Refer to table 32.2,<br>Correspondence   |
|                                 | supply<br>related<br>pins  | HSTBY, RES,<br>NMI, FWE,<br>MD0, MD1,<br>MD2 pins* <sup>4</sup>  | Vin              | -0.3 to 5.5 + 0.3               | V    | - between Power Supply<br>Names and Pins |
|                                 | ASEMD, MD<br>MD3,<br>MD_CLK1,<br>MD_CLK0,<br>MD_CLKP<br>pins* <sup>5</sup> |  | Vin              | $-0.3$ to $V_{cc} + 0.3$        | V    | _  |
|                                 |  | ASEMD, MD4,<br>MD3,<br>MD_CLK1,<br>MD_CLK0,<br>MD_CLKP<br>pins*3 | Vin              | -0.3 to 5.5 + 0.3               | V    | _  |
|                                 | PV <sub>cc</sub> power supply related pins                                 |  | Vin              | -0.3 to PV <sub>cc</sub> + 0.3  | V    | _  |
| Analog su                       | nalog supply voltage   |  | AV <sub>cc</sub> | -0.3 to +6.5                    | V    |  |
| Analog ref                      | Analog reference voltage   |  | AVREFH           | -0.3 to AV <sub>cc</sub> + 0.3  | V    | AVREFH > AVREFL                          |
|                                 |  |  | AVREFL           | $-0.3 \text{ to AV}_{ss} + 0.3$ | V    | _  |
| Analog inp                      | out voltaç   | ge   | V <sub>AN</sub>  | -0.3 to AV <sub>cc</sub> + 0.3  | V    |  |

| Item                                 | Symbol                                | Rating                   | Unit | Remarks           |
|--------------------------------------|---------------------------------------|--------------------------|------|-------------------|
| V <sub>ss</sub> differential voltage | V <sub>ss</sub> – PLLV <sub>s</sub>   | <sub>s</sub> –0.1 to 0.1 | V    | _                 |
|                                      | $V_{SS} - AV_{SS}$                    | -0.1 to 0.1              | V    |                   |
|                                      | PLLV <sub>ss</sub> – AV <sub>ss</sub> | -0.1 to 0.1              | V    |                   |
| Maximum Digital input pins input     | lmax                                  | -25 to +25               | mA   | One pin at a time |
| current per Analog input pins pin    | Imax                                  | -25 to +25               | mA   | _                 |
| Operating temperature*2              | Topr                                  | -40 to +125              | °C   |                   |
| Storage temperature                  | Tstg                                  | -55 to +125              | °C   | Before assembly   |

#### [Operating precautions]

Operating the LSI in excess of the absolute maximum ratings may result in permanent damage. The two power supply voltages of  $PV_{cc}$  of 5V and  $V_{cc}$  of 3.3V may be used simultaneously with the LSI. Be sure to use the LSI in compliance with the connection of power pins, combination conditions of applicable power supply voltages, voltage applicable to each pin, and conditions of output voltage, as specified in the manual. Connecting a non-specified power supply or using the LSI at an incorrect voltage may result in permanent damage of the LSI or the system that contains the LSI.

- Notes: 1. Do not apply any power supply voltage to the V<sub>CL</sub> pin. Connect to GND through an external capacitor.
  - 2. When this LSI is used in the range of 85°C to 125°C, the accumulated operating time must be within 3000 hours.
  - 3. When higher level voltage than  $V_{cc}$  + 0.3 V is input, it is recommended that the voltage input is connected to the pins through the combined resistance of all the pins, 200 k $\Omega$  or higher.
    - However, although the resistance value can be lowered depending on the sink current of external  $V_{\rm cc}$  (3-V system) regulators, the combined resistance value must be 33 k $\Omega$  at least.
    - When the resistance value is modified, make sure that the  $V_{\rm cc}$  voltage does not exceed 3.6 V in the standby state.
  - 4. When the ASEMD, MD4, MD3, MD\_CLK1, MD\_CLK0, and MD\_CLKP pins are input in 5-V tolerant, it is recommended that the HSTBY, RES, NMI, FWE, MD0, MD1, and MD2 pins are input in 5-V amplitude.
  - 5. When the ASEMD, MD4, MD3, MD\_CLK1, MD\_CLK0, and MD\_CLKP pins are input in 3.3-V, the HSTBY, RES, NMI, FWE, MD0, MD1, and MD2 pins can be input in either 3.3-V or 5-V amplitude.

## 32.2 DC Characteristics

Table 32.2 shows the correspondence between power supply names and pins.

Table 32.2 Correspondence between Power Supply Names and Pins

| Pin Namein Input Buffer Driving   |     | Power<br>Supply |         |            | Use | r Pin |            | Power<br>Supply<br>Name | Permissible           |                | Output             |       |
|---|-----|-----------------|---------|------------|-----|-------|------------|-------------------------|-----------------------|----------------|--------------------|-------|
| 5.5 + 0.3*         C14       MD4       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*       Schmitt A 5.5 + 0.3*         C13       MD3       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*       Schmitt A 5.5 + 0.3*         D12       MD2       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         A12       MD1       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         C12       MD0       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         A11       FWE       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         B16       MD_CLK1       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3       Schmitt A         D14       MD_CLK0       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3       Schmitt A         B15       MD_CLKP       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3       Schmitt A         D11       HSTBY       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3       Schmitt A         B12       RES       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         C15       NMI       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         A14       EXTAL       V <sub>cc</sub> V <sub>cc</sub> + 0.3       Schmitt A         A15       XTAL       V <sub>cc</sub> V <sub>cc</sub> + 0.3       Schmitt A         B11       WDTOVF       V <sub>cc</sub> V <sub>cc</sub> + 0.3       Schmitt A         B18       TCK       V <sub>cc</sub> V <sub>cc</sub> + 0.3       TTL         D15       TMS <th></th> <th></th> <th>Symbol</th> <th>Function 1</th> <th></th> <th></th> <th>Function 4</th> <th>-</th> <th>•</th> <th>Buffer<br/>Type</th> <th>Driving<br/>Ability</th> <th>Notes</th>   |     |                 | Symbol  | Function 1 |     |       | Function 4 | -                       | •                     | Buffer<br>Type | Driving<br>Ability | Notes |
| S.5 + 0.3*  | A18 |                 | ASEMD   |            |     |       |            | V <sub>cc</sub>         |                       | Schmitt A      |                    |       |
| D12         MD2         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           A12         MD1         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           C12         MD0         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           A11         FWE         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           B16         MD_CLK1         V <sub>cc</sub> V <sub>cc</sub> + 0.3, Schmitt A         Schmitt A           D14         MD_CLK0         V <sub>cc</sub> V <sub>cc</sub> + 0.3, Schmitt A         Schmitt A           B15         MD_CLKP         V <sub>cc</sub> V <sub>cc</sub> + 0.3, Schmitt A         Schmitt A           D11         HSTBY         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           B12         RES         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           C15         NMI         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           A14         EXTAL         V <sub>cc</sub> V <sub>cc</sub> + 0.3         CMOS           A15         XTAL         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B11         WDTOVF         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B11         WDTOVF         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B15         TMS<   | C14 |                 | MD4     |            |     |       |            | V <sub>cc</sub>         |                       | Schmitt A      |                    |       |
| A12         MD1         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           C12         MD0         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           A11         FWE         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           B16         MD_CLK1         V <sub>cc</sub> V <sub>cc</sub> + 0.3, Schmitt A         Schmitt A           D14         MD_CLK0         V <sub>cc</sub> V <sub>cc</sub> + 0.3, Schmitt A         Schmitt A           B15         MD_CLKP         V <sub>cc</sub> V <sub>cc</sub> + 0.3, Schmitt A         Schmitt A           D11         HSTBY         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           B12         RES         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           C15         NMI         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           A14         EXTAL         V <sub>cc</sub> V <sub>cc</sub> + 0.3         CMOS           A15         XTAL         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B11         WDTOVF         V <sub>cc</sub> B17         TRST         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B18         TCK         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL / Schmitt B  | C13 |                 | MD3     |            |     |       |            | V <sub>cc</sub>         |                       | Schmitt A      |                    |       |
| C12         MD0         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           A11         FWE         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           B16         MD_CLK1         V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*         Schmitt A           D14         MD_CLK0         V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*         Schmitt A           B15         MD_CLKP         V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*         Schmitt A           D11         HSTBY         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           B12         RES         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           C15         NMI         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           A14         EXTAL         V <sub>cc</sub> V <sub>cc</sub> + 0.3         CMOS           A15         XTAL         V <sub>cc</sub> V <sub>cc</sub> + 0.3         CMOS           A13         CK         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B11         WDTOVF         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B18         TCK         V <sub>cc</sub> + 0.3         TTL           D15         TMS         V <sub>cc</sub> + 0.3         TTL           Schmitt B         V <sub>cc</sub> + 0.3         TTL  | D12 |                 | MD2     |            |     |       |            | V <sub>cc</sub>         | 5.5 + 0.3             | Schmitt A      |                    |       |
| A11       FWE       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         B16       MD_CLK1       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*       Schmitt A         D14       MD_CLK0       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*       Schmitt A         B15       MD_CLKP       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*       Schmitt A         D11       HSTBY       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         B12       RES       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         C15       NMI       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         A14       EXTAL       V <sub>cc</sub> V <sub>cc</sub> + 0.3       CMOS         A15       XTAL       V <sub>cc</sub> V <sub>cc</sub> + 0.3       CMOS         B11       WDTOVF       V <sub>cc</sub> V <sub>cc</sub> + 0.3       Schmitt A         B18       TCK       V <sub>cc</sub> V <sub>cc</sub> + 0.3       TTL         D15       TMS       V <sub>cc</sub> V <sub>cc</sub> + 0.3       TTL / Schmitt B   | A12 |                 | MD1     |            |     |       |            | V <sub>cc</sub>         | 5.5 + 0.3             | Schmitt A      |                    |       |
| B16       MD_CLK1       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*       Schmitt A         D14       MD_CLK0       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*       Schmitt A         B15       MD_CLKP       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3       Schmitt A         D11       HSTBY       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         B12       RES       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         C15       NMI       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         A14       EXTAL       V <sub>cc</sub> V <sub>cc</sub> + 0.3       CMOS         A15       XTAL       V <sub>cc</sub> V <sub>cc</sub> + 0.3       CMOS         B11       WDTOVF       V <sub>cc</sub> V <sub>cc</sub> + 0.3       Schmitt A         B18       TCK       V <sub>cc</sub> V <sub>cc</sub> + 0.3       TTL         D15       TMS       V <sub>cc</sub> V <sub>cc</sub> + 0.3       TTL / Schmitt B  | C12 |                 | MD0     |            |     |       |            | V <sub>cc</sub>         | 5.5 + 0.3             | Schmitt A      |                    |       |
| 5.5 + 0.3*         D14       MD_CLK0       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3*       Schmitt A 5.5 + 0.3*         B15       MD_CLKP       V <sub>cc</sub> V <sub>cc</sub> + 0.3, 5.5 + 0.3       Schmitt A 5.5 + 0.3*         D11       HSTBY       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         B12       RES       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         C15       NMI       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         A14       EXTAL       V <sub>cc</sub> V <sub>cc</sub> + 0.3       CMOS         A15       XTAL       V <sub>cc</sub> V <sub>cc</sub> B11       WDTOVF       V <sub>cc</sub> V <sub>cc</sub> + 0.3       Schmitt A         B11       WDTOVF       V <sub>cc</sub> V <sub>cc</sub> + 0.3       Schmitt A         B18       TCK       V <sub>cc</sub> V <sub>cc</sub> + 0.3       TTL         D15       TMS       V <sub>cc</sub> V <sub>cc</sub> + 0.3       TTL / Schmitt B   | A11 |                 | FWE     |            |     |       |            | V <sub>cc</sub>         | 5.5 + 0.3             | Schmitt A      |                    |       |
| B15   MD_CLKP   V <sub>cc</sub>   V <sub>cc</sub> + 0.3,   Schmitt A   5.5 + 0.3*     D11   HSTBY   V <sub>cc</sub>   5.5 + 0.3   Schmitt A     B12   RES   V <sub>cc</sub>   5.5 + 0.3   Schmitt A     C15   NMI   V <sub>cc</sub>   5.5 + 0.3   Schmitt A     A14   EXTAL   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   CMOS     A15   XTAL   V <sub>cc</sub>     A13   CK   V <sub>cc</sub>     B11   WDTOVF   V <sub>cc</sub>     B17   TRST   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   Schmitt A     B18   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL     D15   TMS   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C15   Schmitt B   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C25   TMS   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C36   TMS   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C37   TRST   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   V <sub>cc</sub>   V <sub>cc</sub> + 0.3   TTL / Schmitt B     C38   TCK   TCK | B16 |                 | MD_CLK1 |            |     |       |            | V <sub>cc</sub>         |                       | Schmitt A      |                    |       |
| 5.5 + 0.3*         D11       HSTBY       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         B12       RES       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         C15       NMI       V <sub>cc</sub> 5.5 + 0.3       Schmitt A         A14       EXTAL       V <sub>cc</sub> V <sub>cc</sub> + 0.3       CMOS         A15       XTAL       V <sub>cc</sub> V <sub>cc</sub> A13       CK       V <sub>cc</sub> V <sub>cc</sub> B11       WDTOVF       V <sub>cc</sub> V <sub>cc</sub> + 0.3       Schmitt A         B18       TCK       V <sub>cc</sub> V <sub>cc</sub> + 0.3       TTL         D15       TMS       V <sub>cc</sub> V <sub>cc</sub> + 0.3       TTL / Schmitt B  | D14 |                 | MD_CLK0 | ı          |     |       |            | V <sub>cc</sub>         |                       | Schmitt A      |                    |       |
| B12         RES         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           C15         NMI         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           A14         EXTAL         V <sub>cc</sub> V <sub>cc</sub> + 0.3         CMOS           A15         XTAL         V <sub>cc</sub> A13         CK         V <sub>cc</sub> B11         WDTOVF         V <sub>cc</sub> B17         TRST         V <sub>cc</sub> + 0.3         Schmitt A           B18         TCK         V <sub>cc</sub> + 0.3         TTL           D15         TMS         V <sub>cc</sub> + 0.3         TTL / Schmitt B  | B15 |                 | MD_CLKP |            |     |       |            | V <sub>cc</sub>         |                       | Schmitt A      |                    |       |
| C15         NMI         V <sub>cc</sub> 5.5 + 0.3         Schmitt A           A14         EXTAL         V <sub>cc</sub> V <sub>cc</sub> + 0.3         CMOS           A15         XTAL         V <sub>cc</sub> A13         CK         V <sub>cc</sub> B11         WDTOVF         V <sub>cc</sub> B17         TRST         V <sub>cc</sub> + 0.3         Schmitt A           B18         TCK         V <sub>cc</sub> + 0.3         TTL           D15         TMS         V <sub>cc</sub> + 0.3         TTL / Schmitt B  | D11 |                 | HSTBY   |            |     |       |            | V <sub>cc</sub>         | 5.5 + 0.3             | Schmitt A      |                    |       |
| A14         EXTAL         V <sub>cc</sub> V <sub>cc</sub> + 0.3         CMOS           A15         XTAL         V <sub>cc</sub> V <sub>cc</sub> A13         CK         V <sub>cc</sub> V <sub>cc</sub> B11         WDTOVF         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B17         TRST         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B18         TCK         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           D15         TMS         V <sub>cc</sub> + 0.3         TTL / Schmitt B  | B12 |                 | RES     |            |     |       |            | V <sub>cc</sub>         | 5.5 + 0.3             | Schmitt A      |                    |       |
| A15         XTAL         V <sub>cc</sub> A13         CK         V <sub>cc</sub> B11         WDTOVF         V <sub>cc</sub> B17         TRST         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B18         TCK         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           D15         TMS         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL / Schmitt B   | C15 |                 | NMI     |            |     |       |            | V <sub>cc</sub>         | 5.5 + 0.3             | Schmitt A      |                    |       |
| A13         CK         V <sub>cc</sub> B11         WDTOVF         V <sub>cc</sub> B17         TRST         V <sub>cc</sub> V <sub>cc</sub> + 0.3 Schmitt A           B18         TCK         V <sub>cc</sub> V <sub>cc</sub> + 0.3 TTL           D15         TMS         V <sub>cc</sub> V <sub>cc</sub> + 0.3 TTL / Schmitt B  | A14 |                 | EXTAL   |            |     |       |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3 | CMOS           |                    |       |
| B11         WDTOVF         V <sub>cc</sub> B17         TRST         V <sub>cc</sub> V <sub>cc</sub> + 0.3 Schmitt A           B18         TCK         V <sub>cc</sub> V <sub>cc</sub> + 0.3 TTL           D15         TMS         V <sub>cc</sub> V <sub>cc</sub> + 0.3 TTL / Schmitt B   | A15 |                 | XTAL    |            |     |       |            | V <sub>cc</sub>         |                       |                |                    |       |
| B17         TRST         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           B18         TCK         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           D15         TMS         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL / Schmitt B  | A13 |                 | CK      |            |     |       |            | V <sub>cc</sub>         |                       |                |                    |       |
| $\begin{array}{c ccccccccccccccccccccccccccccccccccc$   | B11 |                 | WDTOVF  |            |     |       |            | V <sub>cc</sub>         |                       |                |                    |       |
| D15 TMS $ V_{\rm cc} = V_{\rm cc} + 0.3 = \frac{TTL}{Schmitt B} $   | B17 |                 | TRST    |            |     |       |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3 | Schmitt A      |                    |       |
| Schmitt B   | B18 |                 | TCK     |            |     |       |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3 | TTL            |                    |       |
| A19 TDI V <sub>cc</sub> V <sub>cc</sub> + 0.3 TTL /   | D15 |                 | TMS     |            |     |       |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3 |                |                    |       |
| Schmitt B   | A19 |                 | TDI     |            |     | _     |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3 |                |                    |       |

| •  |            | Power<br>Supply |         |            | Use        | r Pin      |            | Power<br>Supply<br>Name | Permissible              |                | Output             |       |
|--|------------|-----------------|---------|------------|------------|------------|------------|-------------------------|--------------------------|----------------|--------------------|-------|
| C17         AUDMD         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           D16         AUDRST         V <sub>cc</sub> V <sub>cc</sub> + 0.3         Schmitt A           G18         AUDCK         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           D18         AUDSYNC         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           D17         AUDATA3         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           F18         AUDATA2         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           E17         AUDATA1         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           E18         AUDATA0         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           E19         PA0         A0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           B19         PA1         A1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           E19         PA2         A2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           E19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3 <th>Pin<br/>No.</th> <th>Name<br/>on Pin</th> <th>Symbol</th> <th>Function 1</th> <th>Function 2</th> <th>Function 3</th> <th>Function 4</th> <th>in<br/>Circuit</th> <th>Input<br/>Voltage (V)</th> <th>Buffer<br/>Type</th> <th>Driving<br/>Ability</th> <th>Notes</th> | Pin<br>No. | Name<br>on Pin  | Symbol  | Function 1 | Function 2 | Function 3 | Function 4 | in<br>Circuit           | Input<br>Voltage (V)     | Buffer<br>Type | Driving<br>Ability | Notes |
| D16   AUDRST   | C16        |                 | TDO     |            |            |            |            | V <sub>cc</sub>         |                          |                |                    |       |
| G18         AUDCK         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           D18         ĀUDSYNC         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           D17         AUDATA3         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           F18         AUDATA2         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           E17         AUDATA1         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           E18         AUDATA0         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           C18         PA0         A0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B19         PA1         A1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C19         PA2         A2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G17         PA5         A5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7 <td>C17</td> <td></td> <td>AUDMD</td> <td></td> <td></td> <td></td> <td></td> <td>V<sub>cc</sub></td> <td>V<sub>cc</sub> + 0.3</td> <td>Schmitt A</td> <td></td> <td></td>   | C17        |                 | AUDMD   |            |            |            |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3    | Schmitt A      |                    |       |
| D18         ĀUDSYNC         V <sub>cc</sub> V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           D17         AUDATA3         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           F18         AUDATA2         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           E17         AUDATA1         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           E18         AUDATA0         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           C18         PA0         A0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B19         PA1         A1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C19         PA2         A2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G17         PA5         A5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B  | D16        |                 | AUDRST  |            |            |            |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3    | Schmitt A      |                    |       |
| D17         AUDATA3         V <sub>cc</sub> V <sub>cc</sub> +0.3         TTL           F18         AUDATA2         V <sub>cc</sub> V <sub>cc</sub> +0.3         TTL           E17         AUDATA1         V <sub>cc</sub> V <sub>cc</sub> +0.3         TTL           E18         AUDATA0         V <sub>cc</sub> V <sub>cc</sub> +0.3         TTL           C18         PA0         A0         PV <sub>cc</sub> 1         PV <sub>cc</sub> +0.3         Schmitt B           B19         PA1         A1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           C19         PA2         A2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           E19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           B19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           B19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           B19         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           B19         PA8         A8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           B19  | G18        |                 | AUDCK   |            |            |            |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3    | TTL            |                    |       |
| F18  | D18        |                 | AUDSYNC |            |            |            |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3    | TTL            |                    |       |
| E17         AUDATA1         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           E18         AUDATA0         V <sub>cc</sub> V <sub>cc</sub> + 0.3         TTL           C18         PA0         A0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B19         PA1         A1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C19         PA2         A2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G17         PA5         A5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3   | D17        |                 | AUDATA3 |            |            |            |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3    | TTL            |                    |       |
| E18         AUDATAO         V <sub>cc</sub> V <sub>cc</sub> +0.3         TTL           C18         PA0         A0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           B19         PA1         A1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           C19         PA2         A2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           D19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           G17         PA5         A5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           H18         PA8         A8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3 </td <td>F18</td> <td></td> <td>AUDATA2</td> <td></td> <td></td> <td></td> <td></td> <td>V<sub>cc</sub></td> <td>V<sub>cc</sub> + 0.3</td> <td>TTL</td> <td></td> <td></td>                                     | F18        |                 | AUDATA2 |            |            |            |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3    | TTL            |                    |       |
| C18         PA0         A0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B19         PA1         A1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C19         PA2         A2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G17         PA5         A5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H18         PA8         A8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA12         A12   | E17        |                 | AUDATA1 |            |            |            |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3    | TTL            |                    |       |
| C18         PA0         A0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B19         PA1         A1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C19         PA2         A2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G17         PA5         A5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H18         PA8         A8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA12         A12   | E18        |                 | AUDATA0 |            |            |            |            | V <sub>cc</sub>         | V <sub>cc</sub> + 0.3    | TTL            |                    |       |
| C19         PA2         A2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E19         PA3         A3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G17         PA5         A5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H18         PA8         A8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D20         PA12         A12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15<  | C18        |                 |         | PA0        | A0         |            |            |                         | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| E19         PA3         A3         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           D19         PA4         A4         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           G17         PA5         A5         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           F19         PA6         A6         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           H18         PA8         A8         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           D20         PA12         A12         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>∞</sub> 1         PV <sub>∞</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>∞</sub>   | B19        |                 |         | PA1        | A1         |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| D19         PA4         A4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G17         PA5         A5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H18         PA8         A8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D20         PA12         A12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A  | C19        |                 |         | PA2        | A2         |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| G17         PA5         A5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H18         PA8         A8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D20         PA12         A12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           K18 <t< td=""><td>E19</td><td></td><td></td><td>PA3</td><td>A3</td><td></td><td></td><td>PV<sub>cc</sub>1</td><td>PV<sub>cc</sub>1 + 0.3</td><td>Schmitt B</td><td></td><td></td></t<>                                      | E19        |                 |         | PA3        | A3         |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| F19         PA6         A6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           B20         PA7         A7         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H18         PA8         A8         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           D20         PA12         A12         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B O           K18         PB1         A17         MISOA         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B O   | D19        |                 |         | PA4        | A4         |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| B20         PA7         A7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H18         PA8         A8         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           D20         PA12         A12         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B O           K18         PB1         A17         MISOA         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         Schmitt B O   | G17        |                 |         | PA5        | A5         |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| H18         PA8         A8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           C20         PA9         A9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D20         PA12         A12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O           K18         PB1         A17         MISOA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O  | F19        |                 |         | PA6        | A6         |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| C20         PA9         A9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           J18         PA10         A10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G19         PA11         A11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D20         PA12         A12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O           K18         PB1         A17         MISOA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O   | B20        |                 |         | PA7        | A7         |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| J18       PA10       A10       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       Schmitt B         G19       PA11       A11       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       Schmitt B         D20       PA12       A12       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       Schmitt B         E20       PA13       A13       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       Schmitt B         H19       PA14       A14       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       Schmitt B         F20       PA15       A15       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       Schmitt B         G20       PB0       A16       MOSIA       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       Schmitt B O         K18       PB1       A17       MISOA       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       Schmitt B O  | H18        |                 |         | PA8        | A8         |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| G19         PA11         A11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           D20         PA12         A12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O           K18         PB1         A17         MISOA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O   | C20        |                 |         | PA9        | A9         |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| D20         PA12         A12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           E20         PA13         A13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O           K18         PB1         A17         MISOA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O  | J18        |                 |         | PA10       | A10        |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| E20         PA13         A13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           H19         PA14         A14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O           K18         PB1         A17         MISOA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O   | G19        |                 |         | PA11       | A11        |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| H19         PA14         A14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           F20         PA15         A15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O           K18         PB1         A17         MISOA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O  | D20        |                 |         | PA12       | A12        |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| F20         PA15         A15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           G20         PB0         A16         MOSIA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O           K18         PB1         A17         MISOA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O   | E20        |                 |         | PA13       | A13        |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| G20 PB0 A16 MOSIA PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3 Schmitt B O  K18 PB1 A17 MISOA PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3 Schmitt B O   | H19        |                 |         | PA14       | A14        |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
| K18 PB1 A17 MISOA PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3 Schmitt B O  | F20        |                 |         | PA15       | A15        |            |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      |                    |       |
|  | G20        |                 |         | PB0        | A16        | MOSIA      |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      | 0                  |       |
| J19 PB2 A18 MOSIB PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3 Schmitt B O  | K18        |                 |         | PB1        | A17        | MISOA      |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      | 0                  |       |
|  | J19        |                 |         | PB2        | A18        | MOSIB      |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      | 0                  |       |
| L18 PB3 A19 MISOB PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3 Schmitt B O  | L18        |                 |         | PB3        | A19        | MISOB      |            | PV <sub>cc</sub> 1      | PV <sub>cc</sub> 1 + 0.3 | Schmitt B      | 0                  |       |

| Name   | Pin | Power<br>Supply |        |            | Use        | r Pin      |            | Power<br>Supply<br>Name<br>in | Permissible              |           | Output |       |
|---|-----|-----------------|--------|------------|------------|------------|------------|-------------------------------|--------------------------|-----------|--------|-------|
| J20   |     |                 | Symbol | Function 1 | Function 2 | Function 3 | Function 4 |                               |                          |           |        | Notes |
| NEO   PB6   WEO   PV_c1   PV_c01 + 0.3   Schmitt B  | H20 |                 |        | PB4        | A20        | CTx_B      | TIF6       | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B | 0      |       |
| L17 PB7 WET PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B  K20 PB8 WAIT TOE20 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B  M19 PB9 RD PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B  M19 PB9 RD PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B  M18 PB10 SS0 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B  M20 PB11 SS1 TOE21 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B  M20 PB12 SS2 RSPCKA PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B  N20 PB13 SS3 RSPCKB PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B O  N19 PB13 SS3 RSPCKB PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B  N18 PC0 D0 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B  N18 PC0 D1 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N17 PC2 D2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N17 PC3 D3 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N17 PC3 D3 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC5 D5 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC6 D6 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC6 D6 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC6 D7 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC9 D9 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC9 D9 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC9 D9 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC9 D9 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC9 D9 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC9 D9 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC9 D9 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC9 D9 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N19 PC9 D9 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC11 D11 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC12 D12 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC13 D13 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC14 D14 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC15 D15 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 D12 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC3 D13 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC1 D14 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC1 D12 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC1 D12 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC1 D12 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC1 D12 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC3 D13 D13 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC3 D15 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 TIL  N20 PC2 PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+0.3 Schmitt B | J20 |                 |        | PB5        | A21        | CRx_B      | TIF7       | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B |        |       |
| M19   | K19 |                 |        | PB6        | WE0        |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B |        |       |
| M19         PB9         RD         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         Schmitt B           M18         PB10         CS0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         Schmitt B           M20         PB11         CST         TOE21         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1+ 0.3         Schmitt B           N20         PB12         CS2         RSPCKA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         Schmitt B           N19         PB13         CS3         RSPCKB         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         Schmitt B           N19         PB14         RD/WR         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         Schmitt B           N18         PC0         D0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         TTL           R20         PC1         D1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         TTL           R19         PC2         D2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+ 0.3         TTL           P19  | L17 |                 |        | PB7        | WE1        |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B |        |       |
| M18         PB10         CS0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           M20         PB11         CS1         TOE21         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           N20         PB12         CS2         RSPCKA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B O           N19         PB13         CS3         RSPCKB         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B O           P20         PB14         RD/WR         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           N18         PC0         D0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           R20         PC1         D1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           P19         PC2         D2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           R19         PC4         D4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           R19         PC6   | K20 |                 |        | PB8        | WAIT       | TOE20      |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B |        |       |
| M20         PB11         CSI         TOE21         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         Schmitt B           N20         PB12         CSZ         RSPCKA         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         Schmitt B O           N19         PB13         CSS         RSPCKB         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         Schmitt B O           P20         PB14         RD/WR         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         Schmitt B           N18         PC0         D0         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         TTL           R20         PC1         D1         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         TTL           P19         PC2         D2         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         TTL           N17         PC3         D3         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         TTL           N17         PC3         D3         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         TTL           R19         PC4         D4         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         TTL           R19         PC5         D5         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         TTL           P19         PC6         D6         PV <sub>∞1</sub> 1         PV <sub>∞1</sub> 1 + 0.3         TTL           P19   | M19 |                 |        | PB9        | RD         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B |        |       |
| N20         PB12         CSZ         RSPCKA         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O           N19         PB13         CS3         RSPCKB         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B O           P20         PB14         RD/WR         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           N18         PC0         D0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R20         PC1         D1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC2         D2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           N17         PC4         D4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R19         PC6         D6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           P18         PC7         D7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U19         PC8         D8 <td>M18</td> <td></td> <td></td> <td>PB10</td> <td>CS0</td> <td></td> <td></td> <td>PV<sub>cc</sub>1</td> <td>PV<sub>cc</sub>1 + 0.3</td> <td>Schmitt B</td> <td></td> <td></td>   | M18 |                 |        | PB10       | CS0        |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B |        |       |
| N19         PB13         CS3         RSPCKB         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B O           P20         PB14         RD/WR         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         Schmitt B           N18         PC0         D0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           R20         PC1         D1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           P19         PC2         D2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           T19         PC6         D6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           P18         PC7         D7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           U19         PC9         D9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3         TTL           U19         PC9         D9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1+0.3   | M20 |                 |        | PB11       | CS1        | TOE21      |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B |        |       |
| P20         PB14         RD/WR         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         Schmitt B           N18         PC0         D0         PV <sub>cc</sub> 1 + PV <sub>cc</sub> 1 + 0.3         TTL           R20         PC1         D1         PV <sub>cc</sub> 1 + PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC2         D2         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           T20         PC4         D4         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           T19         PC6         D6         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           P18         PC7         D7         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           U20         PC8         D8         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           U19         PC9         D9         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           V20         PC11         D10         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           V20         PC11         D11         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           V19   | N20 |                 |        | PB12       | CS2        | RSPCKA     |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B | 0      |       |
| N18         PC0         D0         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R20         PC1         D1         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC2         D2         PV <sub>cc</sub> 1 + 0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1 + 0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1 + 0.3         TTL           T20         PC4         D4         PV <sub>cc</sub> 1 + 0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1 + 0.3         TTL           R19         PC6         D6         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC6         D6         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC6         D6         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC6         D6         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC7         D7         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC8         D8         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC9         D9         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC9         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PV  | N19 |                 |        | PB13       | CS3        | RSPCKB     |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B | 0      |       |
| R20         PC1         D1         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           P19         PC2         D2         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           T20         PC4         D4         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           T19         PC6         D6         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           P18         PC7         D7         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           U20         PC8         D8         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           U19         PC9         D9         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           R18         PC10         D10         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           V20         PC11         D11         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           W20         PC12         D12         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           T18         P   | P20 |                 |        | PB14       | RD/WR      |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | Schmitt B |        |       |
| P19         PC2         D2         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           N17         PC3         D3         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T20         PC4         D4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T19         PC6         D6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           P18         PC7         D7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U20         PC8         D8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U19         PC9         D9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R18         PC10         D10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V20         PC11         D11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           W20         PC12         D12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 2 + 0.3 <t< td=""><td>N18</td><td></td><td></td><td>PC0</td><td>D0</td><td></td><td></td><td>PV<sub>cc</sub>1</td><td>PV<sub>cc</sub>1 + 0.3</td><td>TTL</td><td></td><td></td></t<>   | N18 |                 |        | PC0        | D0         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| N17         PC3         D3         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           T20         PC4         D4         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           R19         PC5         D5         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           T19         PC6         D6         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           P18         PC7         D7         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           U20         PC8         D8         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           U19         PC9         D9         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           R18         PC10         D10         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           V20         PC11         D11         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           W20         PC12         D12         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>c0</sub> 1         PV <sub>c0</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>c0</sub> 1         PV <sub>c0</sub> 2 + 0.3         <  | R20 |                 |        | PC1        | D1         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| T20         PC4         D4         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R19         PC5         D5         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T19         PC6         D6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           P18         PC7         D7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U20         PC8         D8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U19         PC9         D9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R18         PC10         D10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V20         PC11         D11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           W20         PC12         D12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>cc</sub> 2         PV <sub>cc</sub> 2 + 0.3         Schmitt B   | P19 |                 |        | PC2        | D2         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| R19       PC5       D5       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         T19       PC6       D6       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         P18       PC7       D7       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         U20       PC8       D8       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         U19       PC9       D9       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         R18       PC10       D10       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         V20       PC11       D11       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         W20       PC12       D12       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         V19       PC13       D13       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         U18       PC14       D14       PV <sub>cc</sub> 1       PV <sub>cc</sub> 1 + 0.3       TTL         T18       PC15       D15       PV <sub>cc</sub> 1       PV <sub>cc</sub> 2 + 0.3       Schmitt B  | N17 |                 |        | PC3        | D3         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| T19         PC6         D6         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           P18         PC7         D7         PV <sub>cc</sub> 1 + 0.3         TTL           U20         PC8         D8         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           U19         PC9         D9         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           R18         PC10         D10         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           V20         PC11         D11         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           W20         PC12         D12         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1 PV <sub>cc</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>cc</sub> 2 PV <sub>cc</sub> 2 + 0.3         Schmitt B  | T20 |                 |        | PC4        | D4         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| P18         PC7         D7         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U20         PC8         D8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U19         PC9         D9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R18         PC10         D10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V20         PC11         D11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           W20         PC12         D12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>cc</sub> 2         PV <sub>cc</sub> 2 + 0.3         Schmitt B  | R19 |                 |        | PC5        | D5         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| U20         PC8         D8         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U19         PC9         D9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R18         PC10         D10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V20         PC11         D11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           W20         PC12         D12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 2 + 0.3         Schmitt B           B6         PD0         TIOC00         TIOC31         PV <sub>cc</sub> 2         PV <sub>cc</sub> 2 + 0.3         Schmitt B  | T19 |                 |        | PC6        | D6         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| U19         PC9         D9         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           R18         PC10         D10         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V20         PC11         D11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           W20         PC12         D12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 2 + 0.3         Schmitt B           B6         PD0         TIOC00         TIOC31         PV <sub>cc</sub> 2         PV <sub>cc</sub> 2 + 0.3         Schmitt B   | P18 |                 |        | PC7        | D7         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| $\begin{array}{c ccccccccccccccccccccccccccccccccccc$   | U20 |                 |        | PC8        | D8         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| V20         PC11         D11         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           W20         PC12         D12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           B6         PD0         TIOC00         TIOC31         PV <sub>cc</sub> 2         PV <sub>cc</sub> 2 + 0.3         Schmitt B   | U19 |                 |        | PC9        | D9         |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| W20         PC12         D12         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           V19         PC13         D13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           B6         PD0         TIOC00         TIOC31         PV <sub>cc</sub> 2         PV <sub>cc</sub> 2 + 0.3         Schmitt B  | R18 |                 |        | PC10       | D10        |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| V19         PC13         D13         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           U18         PC14         D14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           B6         PD0         TIOC00         TIOC31         PV <sub>cc</sub> 2         PV <sub>cc</sub> 2 + 0.3         Schmitt B   | V20 |                 |        | PC11       | D11        |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| U18         PC14         D14         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           T18         PC15         D15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           B6         PD0         TIOC00         TIOC31         PV <sub>cc</sub> 2         PV <sub>cc</sub> 2 + 0.3         Schmitt B  | W20 |                 |        | PC12       | D12        |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| T18         PC15         D15         PV <sub>cc</sub> 1         PV <sub>cc</sub> 1 + 0.3         TTL           B6         PD0         TIOC00         TIOC31         PV <sub>cc</sub> 2         PV <sub>cc</sub> 2 + 0.3         Schmitt B   | V19 |                 |        | PC13       | D13        |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| B6 PD0 TIOC00 TIOC31 PV <sub>cc</sub> 2 PV <sub>cc</sub> 2 + 0.3 Schmitt B  | U18 |                 |        | PC14       | D14        |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
|   | T18 |                 |        | PC15       | D15        |            |            | PV <sub>cc</sub> 1            | PV <sub>cc</sub> 1 + 0.3 | TTL       |        |       |
| B7 PD1 TIOC01 TOE20 PV <sub>cc</sub> 2 PV <sub>cc</sub> 2 + 0.3 Schmitt B   | В6  |                 |        | PD0        | TIOC00     | TIOC31     |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B |        |       |
|   | B7  |                 |        | PD1        | TIOC01     | TOE20      |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B |        |       |

Dower

| Pin        | Power<br>Supply<br>Name |        |            | Use        | r Pin      |            | Power<br>Supply<br>Name<br>in | Permissible<br>Input     | Input<br>Buffer | Output<br>Driving |       |
|------------|-------------------------|--------|------------|------------|------------|------------|-------------------------------|--------------------------|-----------------|-------------------|-------|
| No.        | on Pin                  | Symbol | Function 1 | Function 2 | Function 3 | Function 4 | Circuit                       | Voltage (V)              | Туре            | Ability           | Notes |
| A4         |                         |        | PD2        | TIOC02     | TOE21      | TOE52      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| <b>A</b> 5 |                         |        | PD3        | TIOC03     | TOE22      | TOE53      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| C9         |                         |        | PD4        | TIOC10     | TIOC32     | TOE52      | $PV_{cc}2$                    | $PV_{cc}2 + 0.3$         | Schmitt B       |                   |       |
| B8         |                         |        | PD5        | TIOC11     | TOE23      | TOE40      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| A6         |                         |        | PD6        | TIOC12     |            | TOE41      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| A7         |                         |        | PD7        | TIOC13     |            | TOE42      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| A8         |                         |        | PD8        | TIOC20     | TIOC33     | TOE53      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| В9         |                         |        | PD9        | TIOC21     | TIF0B      | TOE43      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| C10        |                         |        | PD10       | TIOC22     | TIF1B      | TOE50      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| A9         |                         |        | PD11       | TIOC23     | TIF2B      | TOE51      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| D10        |                         |        | PD12       | TCLKA      | TIOC41     | TIJ0       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| C11        |                         |        | PD13       | TCLKB      |            | TIJ1       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| D4         |                         |        | PE0        | TIA00      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| D5         |                         |        | PE1        | TIA01      | TIOC42     | TIOC40     | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| C6         |                         |        | PE2        | TIA02      | TIOC43     | TIOC30     | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| C5         |                         |        | PE3        | TIA03      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| C4         |                         |        | PE4        | TIA04      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| В3         |                         |        | PE5        | TIA05      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| B4         |                         |        | PE6        | TOE00      | CTx_B      |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| D6         |                         |        | PE7        | TOE01      | CRx_B      |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| B5         |                         |        | PE8        | TOE02      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| C7         |                         |        | PE9        | TOE03      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| A2         |                         |        | PE10       | TOE10      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| А3         |                         |        | PE11       | TOE11      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| C8         |                         |        | PE12       | TOE12      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| D8         |                         |        | PE13       | TOE13      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| V11        |                         |        | PF0        | TOD00B     |            | TIF6       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| Y13        |                         |        | PF1        | TOD01B     |            | TIF7       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |

|            | Power<br>Supply |        |            | Haa    | r Pin |            | Power<br>Supply<br>Name | Permissible              | Input          | Output             |       |
|------------|-----------------|--------|------------|--------|-------|------------|-------------------------|--------------------------|----------------|--------------------|-------|
| Pin<br>No. | Name<br>on Pin  | Symbol | Function 1 |        |       | Function 4 | in<br>Circuit           | Input<br>Voltage (V)     | Buffer<br>Type | Driving<br>Ability | Notes |
| Y14        |                 |        | PF2        | TOD02B |       | TIF8       |                         | PV <sub>cc</sub> 2 + 0.3 |                |                    |       |
| W13        |                 |        | PF3        | TOD03B |       | TIF9       | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| V12        |                 |        | PF4        | TOD10B |       | TIF10      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| Y15        |                 |        | PF5        | TOD11B |       | TIF11      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    | -     |
| Y16        |                 |        | PF6        | TOD12B |       | TIF12      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| U13        |                 |        | PF7        | TOD13B |       | TIF13      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| W14        |                 |        | PF8        | TOD20B |       | TIF14      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| Y17        |                 |        | PF9        | TOD21B |       | TIF15      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| V13        |                 |        | PF10       | TOD22B |       | TIF16      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| Y18        |                 |        | PF11       | TOD23B |       | TIF17      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| V14        |                 |        | PF12       | TOD30B |       | TIF18      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| W15        |                 |        | PF13       | TOD31B |       | TIF19      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| Y19        |                 |        | PF14       | TOD32B | CTx_B | TxD_A      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| W16        |                 |        | PF15       | TOD33B | CRx_B | RxD_A      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      |                    |       |
| G1         |                 |        | PG0        | TOD00A | SSLA0 |            | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| НЗ         |                 |        | PG1        | TOD01A | SSLA1 |            | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| H2         |                 |        | PG2        | TOD02A | SSLA2 |            | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| H1         |                 |        | PG3        | TOD03A | SSLA3 |            | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| J1         |                 |        | PG4        | TOD10A | SSLA4 | SSLB3      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| J3         |                 |        | PG5        | TOD11A | SSLA5 | SSLC3      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| J2         |                 |        | PG6        | TOD12A | SSLB0 |            | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| K4         |                 |        | PG7        | TOD13A | SSLB1 |            | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| L1         |                 |        | PG8        | TOD20A | SSLB2 | TIF6       | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| КЗ         |                 |        | PG9        | TOD21A | SSLC0 | TIF7       | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| L2         |                 |        | PG10       | TOD22A | SSLC1 | TIF8       | PV <sub>cc</sub> 2      | $PV_{cc}2 + 0.3$         | Schmitt B      | 0                  |       |
| L3         |                 |        | PG11       | TOD23A | SSLC2 | TIF9       | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| M1         |                 |        | PG12       | TOD30A | SSLA4 | TIF10      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| M4         |                 |        | PG13       | TOD31A | SSLA5 | TIF11      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| M2         |                 |        | PG14       | TOD32A | SSLA6 | TIF12      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |
| N1         |                 |        | PG15       | TOD33A | SSLA7 | TIF13      | PV <sub>cc</sub> 2      | PV <sub>cc</sub> 2 + 0.3 | Schmitt B      | 0                  |       |

| Pin | Power<br>Supply<br>Name |        |            | Use        | r Pin      |                       | Power<br>Supply<br>Name<br>in | Permissible<br>Input     | Input<br>Buffer | Output<br>Driving |       |
|-----|-------------------------|--------|------------|------------|------------|-----------------------|-------------------------------|--------------------------|-----------------|-------------------|-------|
| No. | on Pin                  | Symbol | Function 1 | Function 2 | Function 3 | Function 4            | Circuit                       | Voltage (V)              | Туре            | Ability           | Notes |
| B1  |                         |        | PH0        |            | ADTRG_A    | TIF0A                 | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| E3  |                         |        | PH1        |            | ADTRG_B    | TIF1A                 | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| F4  |                         |        | PH2        |            |            | TIF2A                 | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| B2  |                         |        | PH3        |            |            | TIF3                  | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| D3  |                         |        | PH4        |            |            | TIF4                  | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| СЗ  |                         |        | PH5        |            |            | TIF5                  | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| V18 |                         |        | PJ0        | TxD_A      | CTx_A      | CTx_A &<br>CTx_B      | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| V17 |                         |        | PJ1        | RxD_A      | CRx_A      | CRx_A & CRx_B         | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| V16 |                         |        | PJ2        | TxD_A      | CTx_C      | CTx_A & CTx_B & CTx_C | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| U17 |                         |        | PJ3        | RxD_A      | CRx_C      | CRx_A & CRx_B & CRx_C | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| T17 |                         |        | PJ4        | SCK_A      | ADEND_B    | TIJ0                  | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| V15 |                         |        | PJ5        | TxD_A      |            |                       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| W19 |                         |        | PJ6        | RxD_A      |            |                       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| U15 |                         |        | PJ7        | SCK_B      | ADEND_A    | TIJ1                  | PV <sub>cc</sub> 2            | $PV_{cc}2 + 0.3$         | Schmitt B       | 0                 |       |
| W18 |                         |        | PJ8        | TxD_B      |            |                       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| W17 |                         |        | PJ9        | RxD_B      |            |                       | PV <sub>cc</sub> 2            | $PV_{cc}2 + 0.3$         | Schmitt B       |                   |       |
| G2  |                         |        | PK0        | SCK_C      | RSPCKA     | UBCTRG                | PV <sub>cc</sub> 2            | $PV_{cc}2 + 0.3$         | Schmitt B       | 0                 |       |
| H4  |                         |        | PK1        | TxD_C      | MOSIA      |                       | $PV_{cc}2$                    | $PV_{cc}2 + 0.3$         | Schmitt B       | 0                 |       |
| F1  |                         |        | PK2        | RxD_C      | MISOA      |                       | PV <sub>cc</sub> 2            | $PV_{cc}2 + 0.3$         | Schmitt B       | 0                 |       |
| E1  |                         |        | РК3        | SCK_D      | RSPCKB     |                       | PV <sub>cc</sub> 2            | $PV_{cc}2 + 0.3$         | Schmitt B       | 0                 |       |
| G3  |                         |        | PK4        | TxD_D      | MOSIB      |                       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| F2  |                         |        | PK5        | RxD_D      | MISOB      |                       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| D1  |                         |        | PK6        | SCK_E      |            |                       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| C1  |                         |        | PK7        | TxD_E      |            |                       | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |

| Pin | Power<br>Supply<br>Name |        |            | Use        | r Pin      |            | Power<br>Supply<br>Name<br>in | Permissible<br>Input     | Input<br>Buffer | Output<br>Driving |       |
|-----|-------------------------|--------|------------|------------|------------|------------|-------------------------------|--------------------------|-----------------|-------------------|-------|
| No. | on Pin                  | Symbol | Function 1 | Function 2 | Function 3 | Function 4 | Circuit                       | Voltage (V)              | Туре            | Ability           | Notes |
| E2  |                         |        | PK8        | RxD_E      |            |            | PV <sub>cc</sub> 2            | $PV_{cc}2 + 0.3$         | Schmitt B       |                   |       |
| F3  |                         |        | PK9        |            | RSPCKC     |            | PV <sub>cc</sub> 2            | $PV_{cc}2 + 0.3$         | Schmitt B       | 0                 |       |
| D2  |                         |        | PK10       |            | MOSIC      |            | $PV_{cc}2$                    | $PV_{cc}2 + 0.3$         | Schmitt B       | 0                 |       |
| C2  |                         |        | PK11       |            | MISOC      |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       | 0                 |       |
| P3  |                         |        | PL0        |            | ĪRQ0       |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| P4  |                         |        | PL1        | TOE20      | ĪRQ1       | POD        | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| N3  |                         |        | PL2        | TOE21      | ĪRQ2       |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| R2  |                         |        | PL3        | TOE22      | ĪRQ3       |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| P2  |                         |        | PL4        | TOE23      | ĪRQ4       |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| R1  |                         |        | PL5        | TOE30      | IRQ5       |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| N2  |                         |        | PL6        | TOE31      | ĪRQ6       |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| МЗ  |                         |        | PL7        | TOE32      | ĪRQ7       |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| P1  |                         |        | PL8        | TOE33      |            |            | PV <sub>cc</sub> 2            | PV <sub>cc</sub> 2 + 0.3 | Schmitt B       |                   |       |
| W11 |                         | AN_A0  |            |            |            |            | $AV_{\mathtt{CC}}$            | $AV_{cc} + 0.3$          | Analog          |                   |       |
| U11 |                         | AN_A1  |            |            |            |            | $AV_{\mathtt{cc}}$            | $AV_{cc} + 0.3$          | Analog          |                   |       |
| Y11 |                         | AN_A2  |            |            |            |            | $AV_{\mathtt{cc}}$            | $AV_{cc} + 0.3$          | Analog          |                   |       |
| V10 |                         | AN_A3  |            |            |            |            | $AV_{\mathrm{cc}}$            | $AV_{cc} + 0.3$          | Analog          |                   |       |
| U10 |                         | AN_A4  |            |            |            |            | $AV_{\mathrm{cc}}$            | $AV_{cc} + 0.3$          | Analog          |                   |       |
| Y9  |                         | AN_A5  |            |            |            |            | $AV_{cc}$                     | AV <sub>cc</sub> + 0.3   | Analog          |                   |       |
| V9  |                         | AN_A6  |            |            |            |            | $AV_{\mathrm{cc}}$            | $AV_{cc} + 0.3$          | Analog          |                   |       |
| W9  |                         | AN_A7  |            |            |            |            | $AV_{cc}$                     | AV <sub>cc</sub> + 0.3   | Analog          |                   |       |
| U9  |                         | AN_A8  |            |            |            |            | $AV_{cc}$                     | AV <sub>cc</sub> + 0.3   | Analog          |                   |       |
| Y8  |                         | AN_A9  |            |            |            |            | $AV_{cc}$                     | AV <sub>cc</sub> + 0.3   | Analog          |                   |       |
| Y7  |                         | AN_A10 |            |            |            |            | $AV_{\mathtt{cc}}$            | AV <sub>cc</sub> + 0.3   | Analog          |                   |       |
| Y6  |                         | AN_A11 |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3   | Analog          |                   |       |
| W8  |                         | AN_A12 |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3   | Analog          |                   |       |
| V8  |                         | AN_A13 |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3   | Analog          |                   |       |
| Y5  |                         | AN_A14 |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3   | Analog          |                   |       |

| Pin |                    |          |            |            | r Pin      |            | Power<br>Supply<br>Name<br>in | Permissible<br>Input   | Buffer | Output<br>Driving |       |
|-----|--------------------|----------|------------|------------|------------|------------|-------------------------------|------------------------|--------|-------------------|-------|
|     | on Pin             | Symbol   | Function 1 | Function 2 | Function 3 | Function 4 |                               | Voltage (V)            | Туре   | Ability           | Notes |
| V7  |                    | AN_A15   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| W7  |                    | AN_A16   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| W6  |                    | AN_A17   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| W5  |                    | AN_A18   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| U7  |                    | AN_A19   |            |            |            |            | $AV_{cc}$                     | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| V6  |                    | AN_A20   |            |            |            |            | $AV_{\mathtt{CC}}$            | $AV_{cc} + 0.3$        | Analog |                   |       |
| U6  |                    | AN_A21   |            |            |            |            | $AV_{\mathtt{CC}}$            | $AV_{cc} + 0.3$        | Analog |                   |       |
| V5  |                    | AN_A22   |            |            |            |            | $AV_{cc}$                     | $AV_{cc} + 0.3$        | Analog |                   |       |
| U5  |                    | AN_A23   |            |            |            |            | $AV_{cc}$                     | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| Y2  |                    | AN_A24   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| W2  |                    | AN_A25   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| V3  |                    | AN_A26   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| V4  |                    | AN_A27   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| Y3  |                    | AVREFH_A |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| W3  |                    | AVREFL_A |            |            |            |            | AV <sub>cc</sub>              | AV <sub>ss</sub> + 0.3 | Analog |                   |       |
| T3  |                    | AN_B40   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| U4  |                    | AN_B41   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| U3  |                    | AN_B42   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| T4  |                    | AN_B43   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| R3  |                    | AN_B44   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| V2  |                    | AN_B45   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| R4  |                    | AN_B46   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| U2  |                    | AN_B47   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| T2  |                    | AN_B48   |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| W1  |                    | AVREFH_B |            |            |            |            | AV <sub>cc</sub>              | AV <sub>cc</sub> + 0.3 | Analog |                   |       |
| V1  |                    | AVREFL_B |            |            |            |            | AV <sub>cc</sub>              | AV <sub>ss</sub> + 0.3 | Analog |                   |       |
| A17 | PLLV <sub>cc</sub> |          |            |            |            |            |                               |                        | -      |                   |       |
|     | 30                 |          |            |            |            |            |                               |                        |        |                   |       |

| Pin<br>No. | Power<br>Supply<br>Name<br>on Pin | Symbol   | Function 4 |            | er Pin     | Firmation 4 | Power<br>Supply<br>Name<br>in | Permissible<br>Input<br>Voltage (V) | Buffer | Output<br>Driving<br>Ability | Natas |
|------------|-----------------------------------|----------|------------|------------|------------|-------------|-------------------------------|-------------------------------------|--------|------------------------------|-------|
|            |                                   | Syllibol | runction i | Function 2 | runction 3 | Function 4  | Circuit                       | voitage (v)                         | туре   | Ability                      | Notes |
| A16<br>B10 | PLLV <sub>ss</sub>                |          |            |            |            |             |                               |                                     |        |                              |       |
| K2         | V <sub>CL</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| L19        | V <sub>CL</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| W12        | V <sub>CL</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| B14        |                                   |          |            |            |            |             |                               |                                     |        |                              |       |
| D7         |                                   |          |            |            |            |             |                               |                                     |        |                              |       |
| D13        | V <sub>cc</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| F17        | V <sub>cc</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| G4         | V <sub>cc</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| <br>L4     | V <sub>cc</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| M17        |                                   |          |            |            |            |             |                               |                                     |        |                              |       |
| U8         | V <sub>cc</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| U14        |                                   |          |            |            |            |             |                               |                                     |        |                              |       |
|            | PV <sub>cc</sub> 1                |          |            |            |            |             |                               |                                     |        |                              |       |
| J17        | PV <sub>cc</sub> 1                |          |            |            |            |             |                               |                                     |        |                              |       |
| K17        | PV <sub>cc</sub> 1                |          |            |            |            |             |                               |                                     |        |                              |       |
| P17        | PV <sub>cc</sub> 1                |          |            |            |            |             |                               |                                     |        |                              |       |
|            | PV <sub>cc</sub> 1                |          |            |            |            |             |                               |                                     |        |                              |       |
| D9         | PV <sub>cc</sub> 2                |          |            |            |            |             |                               |                                     |        |                              |       |
| E4         | PV <sub>cc</sub> 2                |          |            |            |            |             |                               |                                     |        |                              |       |
| J4         | PV <sub>cc</sub> 2                |          |            |            |            |             |                               |                                     |        |                              |       |
| N4         | PV <sub>cc</sub> 2                |          |            |            |            |             |                               |                                     |        |                              |       |
| U12        | PV <sub>cc</sub> 2                |          |            |            |            |             |                               |                                     |        |                              |       |
| U16        | PV <sub>cc</sub> 2                |          |            |            |            |             |                               |                                     |        |                              |       |
| A1         | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| A10        | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
| A20        | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                              |       |
|            |                                   |          |            |            |            |             |                               |                                     |        |                              |       |

| Pin<br>No. | Power<br>Supply<br>Name<br>on Pin | Sumah al | Function 4 |            | r Pin      | Franction 4 | Power<br>Supply<br>Name<br>in | Permissible<br>Input<br>Voltage (V) | Buffer | Output<br>Driving | Natas |
|------------|-----------------------------------|----------|------------|------------|------------|-------------|-------------------------------|-------------------------------------|--------|-------------------|-------|
|            |                                   | Symbol   | Function 1 | Function 2 | Function 3 | Function 4  | Circuit                       | voitage (v)                         | туре   | Ability           | Notes |
| B13<br>J9  | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| J10        | V <sub>SS</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| J11        | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| J12        | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
|            | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| K1         | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| K9         | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| K10        |                                   |          |            |            |            |             |                               |                                     |        |                   |       |
| K11<br>K12 | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| L9         | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| L10        | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| L11        | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
|            | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| L20        | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| M9         | V <sub>ss</sub>                   |          |            |            |            |             |                               |                                     |        |                   |       |
| M10        |                                   |          |            |            |            |             |                               |                                     |        |                   |       |
| -          |                                   |          |            |            |            |             |                               |                                     |        |                   |       |
| M11<br>M12 |                                   |          |            |            |            |             |                               |                                     |        |                   |       |
| Y12        |                                   |          |            |            |            |             |                               |                                     |        |                   |       |
| Y20        |                                   |          |            |            |            |             |                               |                                     |        |                   |       |
| T1         | AV <sub>cc</sub>                  |          |            |            |            |             |                               |                                     |        |                   |       |
|            | AV <sub>cc</sub>                  |          |            |            |            |             |                               |                                     |        |                   |       |
| Y4         | AV <sub>cc</sub>                  |          |            |            |            |             |                               |                                     |        |                   |       |
| U1         | AV <sub>ss</sub>                  |          |            |            |            |             |                               |                                     |        |                   |       |
| W4         | AV <sub>ss</sub>                  |          |            |            |            |             |                               |                                     |        |                   |       |
|            | AV <sub>ss</sub>                  |          |            |            |            |             |                               |                                     |        |                   |       |
| Y1         | NC                                |          |            |            |            |             |                               |                                     |        |                   |       |
|            | .10                               |          |            |            |            |             |                               |                                     |        |                   |       |

Note: \* When higher level voltage than  $V_{cc}$  + 0.3 V is input, it is recommended that the voltage input is connected to the pins through the combined resistance of all the pins, 200 k $\Omega$  or higher.

However, although the resistance value can be lowered depending on the sink current of external V $_{\rm cc}$  (3-V system) regulators, the combined resistance value must be 33 k $\Omega$  at least.

When the resistance value is modified, make sure that the  $V_{\rm cc}$  voltage does not exceed 3.6 V in the standby state.

#### [Usage Notes]

Set power supply voltages during LSI operation as shown below.

$$\begin{aligned} &V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, \\ &AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}, AVREFH\_A = AVREFH\_B = 4.5 \text{ V} \text{ to } AV_{cc}, \\ &V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL\_A = AVREFL\_B = 0 \text{ V} \end{aligned}$$

When 
$$PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$$
,  $V_{cc} = PV_{cc}1$ .

The  $PV_{cc}1$  power supply voltage depends on the operating mode as shown below. Operation cannot be guaranteed with other  $PV_{cc}1$  power supply voltages.

Table 32.3 PV<sub>CC</sub>1 Voltage in Each Operating Mode

| Operating |          | Pin Setting |     |     |           |                            |                      |              |  |
|-----------|----------|-------------|-----|-----|-----------|----------------------------|----------------------|--------------|--|
|           | <u> </u> |             | MD1 | MD0 | Mode Name | PV <sub>CC</sub> 1 Voltage |                      |              |  |
| Mode 0    | 0        | 0           | 0*  | 1   | 1         | 1                          | MCU expansion mode   | 3.3 V ±0.3 V |  |
| Mode 1    | 0        | 0           | 1*  | 1   | 1         | 1                          | <del>_</del>         |              |  |
| Mode 2    | 0        | 0           | 0   | 0   | 0         | 1                          | <del>_</del>         |              |  |
| Mode 3    | 0        | 0           | 0   | 0   | 0         | 0                          | MCU single-chip mode | 5.0 V ±0.5 V |  |
| Mode 4    | 0        | 0           | 1   | 0   | 1         | 1                          | Boot mode            | 3.3 V ±0.3 V |  |
| Mode 5    | 0        | 0           | 1   | 0   | 1         | 0                          | <del>_</del>         | 5.0 V ±0.5 V |  |
| Mode 6    | 0        | 0           | 1   | 0   | 0         | 1                          | User program mode    | 3.3 V ±0.3 V |  |
| Mode 7    | 0        | 0           | 1   | 0   | 0         | 0                          | <del>_</del>         | 5.0 V ±0.5 V |  |
| Mode 8    | 0        | 0           | 1   | 1   | 0         | 1                          | User boot mode       | 3.3 V ±0.3 V |  |
| Mode 9    | 0        | 0           | 1   | 1   | 0         | 0                          | _                    | 5.0 V ±0.5 V |  |

Note: \* The FWE input signal determines external bus width when the on-chip ROM disabled MCU expansion mode is selected.

Tables 32.4 to 32.12 show the DC characteristics.

## Table 32.4 DC Characteristics Input Level Voltage

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{\rm cc}$  = 5.0 V ±0.5 V, AVREFH\_A = AVREFH\_B = 4.5 V to  $AV_{\rm cc}$  ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_a = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item  |   | Symbol  | Min.                     | Тур. | Max.                                   | Unit | Measurement<br>Conditions                                 |
|---|---|---|--------------------------|------|--|------|---|
| Schmitt trigger<br>input voltage<br>(buffer type A for<br>mode and<br>control pins) | HSTBY, RES, NMI,<br>FWE, MD2 to MD0                             | V <sub>T</sub> <sup>+</sup> (V <sub>IH</sub> )    | V <sub>cc</sub> × 0.83   | _    | 5.5<br>+ 0.3* <sup>2</sup>             | V    | Refer to table 32.2                                       |
|   | MD_CLK1, MD_CLK0,   | V <sub>T</sub> <sup>+</sup> (V <sub>IH</sub> )    | V <sub>cc</sub> × 0.83   | _    | V <sub>cc</sub><br>+ 0.3* <sup>3</sup> | V    | (pins with<br>Schmitt A input<br>buffers)                 |
| centrer pine)   | MD_CLKP   | V <sub>T</sub> <sup>+</sup> (V <sub>IH</sub> )    | V <sub>cc</sub> × 0.83   | _    | 5.5<br>+ 0.3* <sup>1</sup>             | V    |   |
|   | HSTBY, RES, NMI, ASEMD, FWE, MD4 to                             | · IL  | -0.3                     | _    | V <sub>cc</sub> × 0.2                  | V    |   |
|   | MD0, MD_CLK1,<br>MD_CLK0, MD_CLKP                               | V <sub>HS</sub>                                   | V <sub>cc</sub> × 0.15   | _    | _                                      | V    |   |
|   | TRST, AUDMD,<br>AUDRST  | V <sub>T</sub> <sup>+</sup><br>(V <sub>IH</sub> ) | Vcc<br>× 0.83            | _    | V <sub>cc</sub><br>+ 0.3               | V    |   |
|   |   | V <sub>T</sub> (V <sub>IL</sub> )                 | - 0.3                    | _    | V <sub>cc</sub> × 0.2                  | V    |   |
|   |   | V <sub>HS</sub>                                   | Vcc<br>× 0.15            | _    | _                                      | V    | -   |
| Schmitt trigger input voltage   | PA15 to PA0,<br>PB14 to PB0, PD13 to                            | V <sub>T</sub> <sup>+</sup> (V <sub>IH</sub> )    | PV <sub>cc</sub> × 0.7   | _    | PV <sub>cc</sub><br>+ 0.3              | V    | Refer to table 32.2                                       |
| (buffer type B for GPIO and peripheral IO   | r PD0, PE13 to PE0,<br>PF15 to PF0, PG15 to<br>PG0, PH5 to PH0, | V <sub>T</sub> (V <sub>IL</sub> )                 | -0.3                     | _    | PV <sub>cc</sub> × 0.42                | V    | (pins with<br>Schmitt B<br>input buffers)                 |
| pins)   | PJ9 to PJ0, PK11 to PK0, PL8 to PL0                             | V <sub>HS</sub>                                   | PV <sub>cc</sub> × 0.082 |      | _                                      | V    | -mput bunera)   |
| TTL input<br>voltage (for   | PC15 to PC0   | V <sub>IH</sub>                                   | PV <sub>cc</sub> 1 × 0.7 | _    | PV <sub>cc</sub> 1<br>+ 0.3            | V    | PV <sub>cc</sub> 1 = PV <sub>cc</sub> 2<br>= 5.0 V ±0.5 V |
| GPIO pins)  |   | V <sub>IL</sub>                                   | -0.3                     | _    | PV <sub>cc</sub> 1 × 0.3               | V    | -   |

| Item                          |                                | Symbol   | Min.                     | Тур. | Max.                        | Unit | Measurement Conditions                              |
|-------------------------------|--------------------------------|--|--------------------------|------|-----------------------------|------|---|
| TTL input voltage (for        | D15 to D0 (MCU expansion mode) | V <sub>IH</sub>                                | 2.2                      | _    | PV <sub>cc</sub> 1<br>+ 0.3 | V    | PV <sub>cc</sub> 1 = V <sub>cc</sub> = 3.3 V ±0.3 V |
| extended data bus pins)       |                                | V <sub>IL</sub>                                | -0.3                     | _    | 8.0                         | V    |   |
| Clock input pin voltage       | EXTAL                          | V <sub>IH</sub>                                | V <sub>cc</sub><br>× 0.7 | _    | V <sub>cc</sub><br>+ 0.3    | V    |   |
|                               |                                | V <sub>IL</sub>                                | -0.3                     | _    | V <sub>cc</sub><br>× 0.2    | V    |   |
| TTL input<br>voltage (for     | AUDCK, AUDSYNC,<br>AUDATA3 to  | V <sub>IH</sub>                                | 2.2                      | _    | V <sub>cc</sub><br>+ 0.3    | V    |   |
| AUD-II and<br>H-UDI)          | AUDATA0,<br>TCK, TMS, TDI      | V <sub>IL</sub>                                | -0.3                     | _    | 0.8                         | V    |   |
| Schmitt trigger input voltage | TMS, TDI                       | V <sub>T</sub> <sup>+</sup> (V <sub>IH</sub> ) | V <sub>cc</sub><br>× 0.7 | _    | V <sub>cc</sub><br>+ 0.3    | V    | Refer to table 32.2                                 |
| (buffer type B for JTAG)      | •                              | V <sub>⊤</sub> - (V <sub>I</sub> L)            | -0.3                     | _    | V <sub>cc</sub><br>× 0.42   | V    | ⁻(pins with<br>Schmitt B input<br>₋buffers)         |
|                               |                                | V <sub>HS</sub>                                | V <sub>cc</sub> × 0.082  |      | _                           | V    | ,   |

Notes: 1. When higher level voltage than  $V_{cc}$  + 0.3 V is input, it is recommended that the voltage input is connected to the pins through the combined resistance of all the pins, 200 k $\Omega$  or higher.

However, although the resistance value can be lowered depending on the sink current of external V $_{\rm cc}$  (3-V system) regulators, the combined resistance value must be 33 k $\Omega$  at least.

When the resistance value is modified, make sure that the  $\rm V_{cc}$  voltage does not exceed 3.6 V in the standby state.

- 2. When the ASEMD, MD4, MD3, MD\_CLK1, MD\_CLK0, and MD\_CLKP pins are input in 5-V tolerant, it is recommended that the HSTBY, RES, NMI, FWE, MD0, MD1, and MD2 pins are input in 5-V amplitude.
- 3. When the ASEMD, MD4, MD3, MD\_CLK1, MD\_CLK0, and MD\_CLKP pins are input in 3.3-V, the HSTBY, RES, NMI, FWE, MD0, MD1, and MD2 pins can be input in either 3.3-V or 5-V amplitude.

#### Table 32.5 DC Characteristics Input Leak Current

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc}$  = 5.0 V ±0.5 V,  $AVREFH_A$  =  $AVREFH_B$  = 4.5 V to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_a = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item                  |   | Symbol  | Min. | Тур. | Max. | Unit | Measurement<br>Conditions  |
|-----------------------|---|---------|------|------|------|------|--|
| Input leak<br>current | HSTBY, RES, NMI, ASEMD, FWE, MD4 to MD0, MD_CLK1, MD_CLK0, MD_CLKP, EXTAL                                       | l lin l | _    | _    | 2.0  | μΑ   | $Vin = 0.3 V to$ $V_{cc} - 0.3 V$  |
|                       | TRST, TMS, TDI, TCK, AUDMD, AUDRST, AUDCK, AUDSYNC, AUDATA3 to AUDATA0 (when pull-up/down resistor is off)      | _       |      |      | 2.0  | μΑ   | $Vin = 0.3 V to$ $V_{cc} - 0.3 V$  |
|                       | PA15 to PA0, PB14 to PB0 (extended bus mode)  | _       | _    | _    | 2.0  | μΑ   | Vin = 0.3 V to<br>$PV_{cc}1 - 0.3 V$ ,<br>$PV_{cc}1 = V_{cc} = 3.3$<br>$V \pm 0.3 V$   |
|                       | PC15 to PC0 (D15 to D0) (standby in expansion bus mode)   | _       | _    | _    | 2.0  | μΑ   | Vin = 0.3 V to PVcc1 - 0.3 V, PVcc1 = Vcc = 3.3 V ± 0.3 V                              |
|                       | PA15 to PA0, PB14 to<br>PB0, PC15 to PC0 (other<br>than extended bus mode)                                      | _       | _    | _    | 2.0  | μΑ   | Vin = 0.3 V to<br>$PV_{cc}1 - 0.3 V$ ,<br>$PV_{cc}1 = PV_{cc}2 =$<br>$5.0 V \pm 0.5 V$ |
|                       | PD13 to PD0,<br>PE13 to PE0, PF15 to PF0,<br>PG15 to PG0, PH5 to PH0,<br>PJ9 to PJ0, PK11 to PK0,<br>PL8 to PL0 |         | _    | _    | 2.0  | μΑ   | Vin = 0.3  V to<br>$PV_{cc}2 - 0.3 \text{ V}$  |

| Item                  |          | Symbol  | Min. | Тур. | Max. | Unit | Measurement Conditions   |
|-----------------------|----------|---------|------|------|------|------|--|
| Input leak<br>current | A/D port | l lin l | _    | _    | 0.1  | μΑ   | Vin = 0.3 V to<br>$AV_{cc} - 0.3 V$ ,<br>$T_a = -40^{\circ}C$ to<br>$105^{\circ}C$ |
|                       |          |         | _    | _    | 0.2  | μΑ   | Vin = 0.3 V to<br>$AV_{cc} - 0.3 V$ ,<br>$T_a = 105^{\circ}C$ to<br>$125^{\circ}C$ |

#### Table 32.6 DC Characteristics Pull-Up/Pull-Down MOS Current

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc} = AV_{cc} = 4.5 \text{ V} =$$

| Item                         |  | Symbol | Min. | Тур. | Max. | Unit | Measurement Conditions |
|------------------------------|--|--------|------|------|------|------|------------------------|
| Input pull-up<br>MOS current | TMS, TRST, TDI,<br>TCK                             | –lpu   | _    | _    | 350  | μА   | Vin = 0 V              |
|                              | AUDMD, AUDCK,<br>AUDSYNC,<br>AUDATA3 to<br>AUDATA0 | _      | _    | _    | 350  | μΑ   | Vin = 0 V              |
|                              | D15 to D0 (MCU expansion mode)                     | _      |      | _    | 160  | μА   | Vin = 0 V              |
| Input pull-down              | AUDRST   | lpd    | _    | _    | 350  | μА   | Vin = V <sub>cc</sub>  |
| MOS current                  | RxD_A to RxD_E                                     | _      | _    | _    | 350  | μА   | Vin = PV <sub>cc</sub> |
|                              | MISOA to MISOC                                     | _      | _    | _    | 350  | μА   | Vin = PV <sub>cc</sub> |
|                              | ASEMD  | _      | _    | _    | 200  | μА   | Vin = V <sub>cc</sub>  |

#### Table 32.7 DC Characteristics — Output Level Voltage

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH_A = AVREFH_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL\_A = AVREFL\_B = 0 V,$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_a = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item                            |  | Symbol          | Min.                        | Тур. | Max. | Unit | Measurement<br>Conditions   |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |   |   |  |   |   |   |                       |                   |                       |   |                          |   |                          |
|---------------------------------|--|-----------------|-----------------------------|------|------|------|---|--|--|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|---|---|--|---|---|---|-----------------------|-------------------|-----------------------|---|--------------------------|---|--------------------------|
| Output<br>high-level<br>voltage | PA15 to PA0, PB14,<br>PB13, PB11 to PB1,<br>PC15 to PC0 (MCU<br>expansion mode)  | V <sub>OH</sub> | PV <sub>cc</sub> 1<br>- 0.5 | _    | _    | V    | $\begin{split} I_{_{OH}} &= 200 \; \mu A, \\ PV_{_{CC}} 1 &= V_{_{CC}} = 3.3 \; V \\ &\pm 0.3 \; V \end{split}$ |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |   |   |  |   |   |   |                       |                   |                       |   |                          |   |                          |
|                                 | PA15 to PA0, PB14,<br>PB13, PB11 to PB1,<br>PC15 to PC0 (When  | <del>-</del>    | PV <sub>cc</sub> 1<br>- 0.5 | _    | _    | V    | $I_{OH} = 200 \mu A,$<br>$PV_{CC}1 = PV_{CC}2 =$<br>$5.0 \text{ V} \pm 0.5 \text{ V}$                           |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |   |   |  |   |   |   |                       |                   |                       |   |                          |   |                          |
|                                 | not in MCU expansion mode)   |                 | PV <sub>cc</sub> 1<br>- 1.0 | _    | _    | V    | $I_{OH} = 1 \text{ mA},$<br>$PV_{CC}1 = PV_{CC}2 =$<br>$5.0 \text{ V} \pm 0.5 \text{ V}$                        |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |   |   |  |   |   |   |                       |                   |                       |   |                          |   |                          |
|                                 | PD13 to PD0, PE13<br>to PE0, PF15, PF13  | _               | PV <sub>cc</sub> 2<br>- 0.5 | _    | _    | V    | I <sub>OH</sub> = 200 μA  |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |   |   |  |   |   |   |                       |                   |                       |   |                          |   |                          |
|                                 | to PF0, PG15 to<br>PG0, PH5 to PH0,<br>PJ9, PJ8, PJ6 to PJ1,<br>PK11 to PK0, PL8 to<br>PL0, PJ0, PB0,<br>PF14, PJ7, PB12 |                 | PV <sub>cc</sub> 2<br>- 1.0 | _    | _    | V    | I <sub>OH</sub> = 1 mA  |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |   |   |  |   |   |   |                       |                   |                       |   |                          |   |                          |
|                                 | CK   | -               | V <sub>cc</sub> – 0.5       | _    | _    | V    | I <sub>OH</sub> = 200 μA  |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |   |   |  |   |   |   |                       |                   |                       |   |                          |   |                          |
|                                 | WDTOVF   |                 | -                           | -    | -    | -    | -   |  |  |  | - | - | - | - | - | - | - | - | - | - | _ | _ | _ | _ | - | - | - | _ | - | _ | _ | - | - | - | - | _ | - | -<br>- | _ | _ |  | _ | _ | _ | _                     | <del>-</del><br>- | V <sub>cc</sub> – 0.5 |   | _                        | V | Ι <sub>οн</sub> = 200 μΑ |
|                                 | TDO  |                 |                             |      |      |      |   |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |   |   |  |   |   |   | V <sub>cc</sub> – 0.5 | _                 | _                     | V | I <sub>OH</sub> = 200 μA |   |                          |
|                                 | AUDCK, AUDSYNC,<br>AUDATA3 to<br>AUDATA0   | _               | V <sub>cc</sub> – 0.5       | _    | _    | V    | Ι <sub>οн</sub> = 200 μΑ  |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |        |   |   |  |   |   |   |                       |                   |                       |   |                          |   |                          |

| Item                            |  | Symbol          | Min. | Тур. | Max. | Unit | Measurement Conditions   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                   |   |                   |              |          |                   |              |        |                   |                   |          |        |   |                   |        |   |          |        |          |   |   |     |   |                        |
|---------------------------------|--|-----------------|------|------|------|------|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------------------|---|-------------------|--------------|----------|-------------------|--------------|--------|-------------------|-------------------|----------|--------|---|-------------------|--------|---|----------|--------|----------|---|---|-----|---|------------------------|
| Output low-<br>level<br>voltage | PA15 to PA0, PB14,<br>PB13, PB11 to PB1,<br>PC15 to PC0 (MCU<br>expansion mode)  | V <sub>oL</sub> | _    |      | 0.4  | V    | $I_{oL} = 1.6 \text{ mA},$<br>$PV_{cc}1 = V_{cc} = 3.3 \text{ V}$<br>$\pm 0.3 \text{ V}$   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                   |   |                   |              |          |                   |              |        |                   |                   |          |        |   |                   |        |   |          |        |          |   |   |     |   |                        |
|                                 | PA15 to PA0, PB14,<br>PB13, PB11 to PB1,<br>PC15 to PC0 (other   | <del>-</del>    | _    | _    | 0.4  | V    | $I_{oL} = 1.6 \text{ mA},$<br>$PV_{cc}1 = PV_{cc}2 =$<br>$5.0 \text{ V} \pm 0.5 \text{ V}$ |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                   |   |                   |              |          |                   |              |        |                   |                   |          |        |   |                   |        |   |          |        |          |   |   |     |   |                        |
|                                 | than MCU expansion mode)   |                 | _    | _    | 1.2  | V    | $I_{oL} = 4 \text{ mA},$<br>$PV_{cc}1 = PV_{cc}2 =$<br>$5.0 \text{ V} \pm 0.5 \text{ V}$   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                   |   |                   |              |          |                   |              |        |                   |                   |          |        |   |                   |        |   |          |        |          |   |   |     |   |                        |
|                                 | PD13 to PD0, PE13  | =               | _    | _    | 0.4  | V    | I <sub>OL</sub> = 1.6 mA   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                   |   |                   |              |          |                   |              |        |                   |                   |          |        |   |                   |        |   |          |        |          |   |   |     |   |                        |
|                                 | to PE0, PF15, PF13<br>to PF0, PG15 to<br>PG0, PH5 to PH0,<br>PJ9, PJ8, PJ6 to PJ1,<br>PK11 to PK0, PL8 to<br>PL0, PJ0, PB0,<br>PF14, PJ7, PB12 |                 |      | _    | 1.2  | V    | I <sub>oL</sub> = 4 mA   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                   |   |                   |              |          |                   |              |        |                   |                   |          |        |   |                   |        |   |          |        |          |   |   |     |   |                        |
|                                 | СК   | _               | _    |      | 0.4  | V    | I <sub>OL</sub> = 1.6 mA   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                   |   |                   |              |          |                   |              |        |                   |                   |          |        |   |                   |        |   |          |        |          |   |   |     |   |                        |
|                                 | WDTOVF   | -               | -    | =    | -    | -    | =  | = | - | - | _ | _ | _ | _ | _ | _ | _ | _ | - | - | _ | _ | _ | _ | _ | <del>-</del><br>- | _ | <del>-</del><br>- | <del>-</del> | <u>-</u> | <del>-</del><br>- | <del>-</del> | _<br>_ | <del>-</del><br>- | <del>-</del><br>- | <u>-</u> | _<br>_ | _ | <del>-</del><br>- | _<br>_ | _ | <u> </u> | _<br>_ | <u> </u> | _ | _ | 0.4 | V | I <sub>OL</sub> = 1 mA |
|                                 | TDO  | _               | _    | _    | 0.4  | V    | I <sub>OL</sub> = 1.6 mA   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                   |   |                   |              |          |                   |              |        |                   |                   |          |        |   |                   |        |   |          |        |          |   |   |     |   |                        |
|                                 | AUDCK, AUDSYNC,<br>AUDATA3 to<br>AUDATA0   |                 | _    | _    | 0.4  | V    | I <sub>oL</sub> = 1.6 mA   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                   |   |                   |              |          |                   |              |        |                   |                   |          |        |   |                   |        |   |          |        |          |   |   |     |   |                        |

#### Table 32.8 DC Characteristics — Permissible Output Current Values

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{CC}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_{0} = -40^{\circ}\text{C} \text{ to } 125^{\circ}\text{C}$ 

| Item  | Symbol                   | Min. | Тур. | Max. | Unit |
|---|--------------------------|------|------|------|------|
| Output low-level permissible current (per pin)  | I <sub>OL</sub>          | —    | _    | 4.0  | mA   |
| Output low-level permissible current (total)    | $\Sigma$ $I_{OL}$        | _    | _    | 80   | mA   |
| Output high-level permissible current (per pin) | I <sub>OH</sub>          | _    | _    | 2.0  | mA   |
| Output high-level permissible current (total)   | $\Sigma$ I <sub>OH</sub> | _    | _    | 25   | mA   |

#### [Operating precautions]

To assure LSI reliability, do not exceed the output values listed in this table.

## Table 32.9 DC Characteristics — Injection Current Values

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V} = 3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{CC}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL\_A = AVREFL\_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_a = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item                         |            | Symbol                    | Min. | Тур. | Max. | Unit |
|------------------------------|------------|---------------------------|------|------|------|------|
| DC Injection current         | logic pin  | I <sub>IC</sub>           | -1.0 | _    | 2.0  | mA   |
| (per pin)                    | Analog pin |                           | -3.0 | _    | 3.0  | mA   |
| DC Injection current (total) |            | $\Sigma \mid I_{IC} \mid$ | _    | _    | 50.0 | mA   |

Note: Make sure that the voltages for the pins do not exceed 5.8 V.

## Table 32.10 DC Characteristics — Input Capacitance

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH_A = AVREFH_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

| Item              |                | Symbol | Min. | Тур. | Max. | Unit | Measurement<br>Conditions                         |
|-------------------|----------------|--------|------|------|------|------|---|
| Input capacitance | All input pins | Cin    | _    | 10   | 20   | pF   | Vin = 0 V,<br>f = 1 MHz,<br>T <sub>a</sub> = 25°C |

## Table 32.11 DC Characteristics — Supply Current

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{\rm cc}$  = 5.0 V ±0.5 V, AVREFH\_A = AVREFH\_B = 4.5 V to  $AV_{\rm cc}$  ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_a = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item   |                                 | Symbol            | Min. | Тур.* | Max. | Unit | Measurement Conditions |
|--|---------------------------------|-------------------|------|-------|------|------|------------------------|
| Supply current                                       | Normal operation                | I <sub>cc</sub>   | _    | 99    | 165  | mA   | f = 80 MHz             |
| (V <sub>cc</sub> power supply)                       | (Including flash                |                   | _    | 122   | 188  | mA   | f = 120 MHz            |
|  | memory programming and erasure) |                   | _    | 145   | 211  | mA   | f = 160 MHz            |
|  |                                 |                   | _    | 169   | 234  | mA   | f = 200 MHz            |
|  | Sleep                           | I <sub>SLP</sub>  |      | 53    | 112  | mΑ   | f = 80 MHz             |
|  |                                 |                   | _    | 59    | 118  | mA   | f = 120 MHz            |
|  |                                 |                   | _    | 65    | 124  | mA   | f = 160 MHz            |
|  |                                 |                   | _    | 70    | 130  | mA   | f = 200 MHz            |
|  | At reset                        | IRST              | 30   | 66    | 129  | mA   | f = 80 MHz             |
|  |                                 |                   | 40   | 75    | 138  | mA   | f = 120 MHz            |
|  |                                 |                   | 50   | 85    | 148  | mA   | f = 160 MHz            |
|  |                                 |                   | 60   | 94    | 157  | mA   | f = 200 MHz            |
| PLL supply current (PLLV <sub>cc</sub> power supply) |                                 | I <sub>PLL</sub>  | _    | 4.0   | 7.5  | mA   |                        |
| Analog supply current (AV <sub>cc</sub> power        | During A/D conversion           | I <sub>AVCC</sub> | _    | 7.4   | 9.8  | mA   | for 2 modules          |
| supply)  | Awaiting A/D conversion         | _                 | _    | 65    | 120  | μА   | -                      |
|  | Standby                         | =                 | _    | 5.0   | 100  | μΑ   | _                      |
| ADC reference power supply                           | During A/D conversions          | AVREF             | _    | 2.5   | 4.0  | mA   | for 2 modules          |
| current (AVREF)                                      | Awaiting A/D conversion         | -                 | _    | 2.2   | 3.5  | mA   | -                      |
|  | Standby                         | -                 | _    | 0.1   | 1.0  | μΑ   | _                      |

Note: \* The Typ condition is  $V_{cc} = 3.3V$ ,  $T_a = 25$  °C.

Apr 01, 2014

#### [Operating precautions]

- 1. When the A/D converter is not used (including during standby), do not leave the  $AV_{CC}$ ,  $AV_{ref}$ , and AV<sub>ss</sub> pins open.
- 2. The supply current is measured when  $V_{H}$ min =  $V_{CC} 0.5 \text{ V/PV}_{CC} 0.5 \text{ V}$ ,  $V_{IL} = 0.5 \text{ V}$ , with all output pins unloaded.
- 3. The guaranteed operating range of power supply PV<sub>cc</sub>1 in MCU expansion modes is only  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ . Do not use a voltage outside this range.
- 4. The guaranteed operating range of power supply PV<sub>cc</sub>1 in MCU single-chip mode is only  $PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}$ . Do not use a voltage outside this range.

#### Table 32.12 DC Characteristics — Standby

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, PV_{cc}2 = 4.5 \text{ V} + 4.5 \text{ V}$$

| Item   |                    | Symbol           | Min. | Тур. | Max. | Unit | Measurement Conditions |
|--|--------------------|------------------|------|------|------|------|------------------------|
| Supply current (V <sub>cc</sub> power supply)      | Standby            | I <sub>SB</sub>  |      | _    | 300  | μА   | -40°C ≤ Ta<br>≤ 50°C   |
|  |                    |                  | _    | _    | 750  | μА   | 50°C < Ta<br>≤ 105°C   |
|  |                    |                  |      | _    | 1000 | μА   | 105°C < Ta<br>≤ 125°C  |
| RAM standby voltage (V <sub>cc</sub> power supply) | RAM data retention | V <sub>RAM</sub> | 2.7  | _    | _    | V    |                        |

#### 32.3 **AC Characteristics**

#### 32.3.1 **Timing for Power On and Off**

Table 32.13 shows the timing for power on and off.

#### Table 32.13 Timing for Power On and Off

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$$

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_{a} = -40^{\circ}\text{C} \text{ to } 125^{\circ}\text{C}$ 

| Item                              | Symbol             | Min. | Max. | Unit | Figures     |
|-----------------------------------|--------------------|------|------|------|-------------|
| Preceding Vcc power-on time       | t <sub>vccs</sub>  | 0    | _    | ms   | Figure 32.1 |
| Vcc holding time at PVcc shutdown | t <sub>vcch1</sub> | 0    | _    | ms   | _           |

If  $t_{vccs}$  or  $t_{vcch}$  are not satisfied, status of pins operating on PVcc1 or PVcc2 cannot be Note: guaranteed.

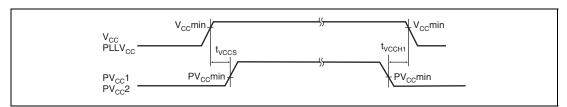


Figure 32.1 Timing for Power On and Off

## 32.3.2 Timing for Operation Mode and Oscillation

Table 32.14 shows the timing for operation mode and oscillation.

#### Table 32.14 Timing for operation mode and oscillation

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}1 = 4.5 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 4.5 \text{ V}, AV_{cc$$

| Item   |                     | Symbol             | Min. | Max. | Unit             | Figures     |
|--|---------------------|--------------------|------|------|------------------|-------------|
| Oscillation settling time  |                     | t <sub>osc1</sub>  | 10   | _    | ms               | Figure 32.2 |
| Operation mode set up time when sta  | t <sub>MDS1</sub>   | 10                 | _    | ms   | _                |             |
| Operation mode set up time during  | MD_CLKP = 0         | t <sub>MDS2</sub>  | 10   | _    | t <sub>cyc</sub> | _           |
| operation  | MD_CLKP = 1         | _                  | 20   | _    | _                |             |
| Operation mode hold time after reset   | MD_CLKP = 0         | t <sub>MDH1</sub>  | 30   | _    | t <sub>cyc</sub> | _           |
| is inactive  | MD_CLKP = 1         | _                  | 60   | _    | _                |             |
| Operation mode hold time when pow  | er down             | t <sub>MDH2</sub>  | 0    | _    | ms               | _           |
| V <sub>cc</sub> hold time when switched off. (Avoid damage during flash programm | ming or erasing)    | t <sub>vcch2</sub> | 22   | _    | μS               | _           |
| HSTBY hold time after reset (Keep RAM contents during reset sta                  | t <sub>HSTBYH</sub> | 1                  | _    | μS   | _                |             |
| HSTBY pulse width (Power down and re-start internal reg                          | t <sub>HSTBYW</sub> | 1                  | _    | ms   | Figure 32.3      |             |

Note: AVREFH\_A ≤ AVcc + 0.3 V and AVREFH\_B ≤ AVcc + 0.3 V must always be satisfied even at power-on/off.

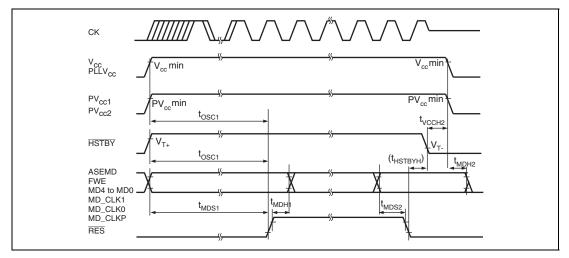


Figure 32.2 Operation Mode and Oscillation Timing when Power-On/Off

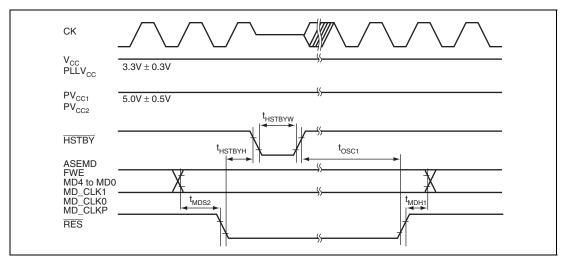


Figure 32.3 Operation Mode and Oscillation Timing during Operation

## 32.3.3 Clock Timing

Table 32.15 shows the clock timing.

## Table 32.15 Clock Timing

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}1 = 4.5 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 4.5 \text{ V} \pm 0.5 \text{ V}, AV_{cc}3 = 4.5 \text{ V} \pm 0.5 \text{ V}, AV_{cc}4 = 4.5 \text{ V}, AV_{cc}4 = 4.5 \text{ V}, AV_{cc}4 = 4.5 \text{$$

| Item                          |             | Symbol           | Min. | Max.  | Unit | Figures     |
|-------------------------------|-------------|------------------|------|-------|------|-------------|
| EXTAL clock input frequency   |             | f <sub>ex</sub>  | 16   | 20    | MHz  | Figure 32.4 |
| EXTAL clock input cycle time  |             | t                | 50   | 62.5  | ns   | _           |
| EXTAL clock input low-level p | oulse width | t <sub>ext</sub> | 15   | _     | ns   | _           |
| EXTAL clock input high-level  | pulse width | t <sub>exh</sub> | 15   | _     | ns   | _           |
| EXTAL clock input rise time   |             | t <sub>exr</sub> | _    | 4     | ns   | _           |
| EXTAL clock input fall time   |             | t                | _    | 4     | ns   | _           |
| Clock frequency*1             | MD_CLKP = 0 | f <sub>op</sub>  | 16   | 20    | MHz  | Figure 32.5 |
| Clock cycle time              | _           | t <sub>cyc</sub> | 50   | 62.5  | ns   |             |
| Clock low-level pulse width   | _           | t <sub>cl</sub>  | 12   | _     | ns   | _           |
| Clock high-level pulse width  | _           | t <sub>ch</sub>  | 12   | _     | ns   | _           |
| Clock frequency*1             | MD_CLKP = 1 | f <sub>op</sub>  | 32   | 40    | MHz  | _           |
| Clock cycle time              | _           | t <sub>cyc</sub> | 25   | 31.25 | ns   |             |
| Clock low-level pulse width   | _           | t <sub>cl</sub>  | 4    | _     | ns   | _           |
| Clock high-level pulse width  | _           | t <sub>ch</sub>  | 4    | _     | ns   | _           |
| Clock rise time               |             | t <sub>cr</sub>  | _    | 8     | ns   |             |
| Clock fall time               |             | t <sub>CF</sub>  |      | 8     | ns   |             |
| Clock jitter (averaged 1µs)*2 |             | f <sub>JIT</sub> |      | 0.3   | %    | Not tested  |

Notes: 1. The CK pin outputs the peripheral clock signal ( $P\phi$ ).

2. The clock jitter depends on board design.

Page 1737 of 1824

# [Operating precautions]

The EXTAL, XTAL, and CK pins constitute a circuit requiring a power supply voltage of  $V_{cc}$  = 3.3 V ±0.3 V. Comply with the input and output voltages specified in the DC characteristics.

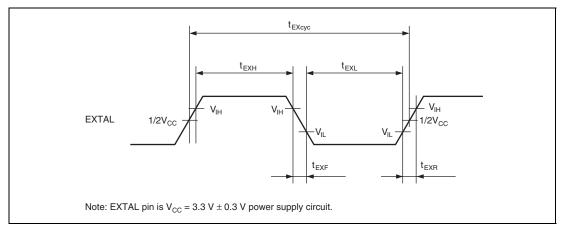


Figure 32.4 EXTAL Clock Input Timing

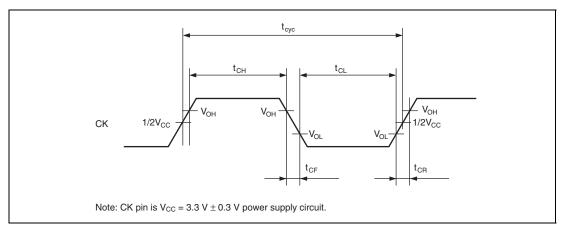


Figure 32.5 Peripheral Clock Timing

## 32.3.4 Control Signal Timing

Table 32.16 shows the control signal timing.

#### **Table 32.16 Control Signal Timing**

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AVREFH_A = AVREFH_B = 4.5 \text{ V} \text{ to } AV_{cc}, AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}, AVREFH_A = AVREFH_B = 4.5 \text{ V} \text{ to } AV_{cc}, AV_{cc} = PLLV_{cc} = AV_{cc} = AV_{c$$

| Item   |                    | Symbol              | Min. | Max. | Unit             | Figures     |
|--|--------------------|---------------------|------|------|------------------|-------------|
| RES pulse width (Except flash programming or   | MD_CLKP = 0        | t <sub>RESW1</sub>  | 30   | _    | t <sub>cyc</sub> | Figure 32.6 |
| erasing)                                       | MD_CLKP = 1        | _                   | 60   | _    | _                |             |
| RES pulse width (During flash programming or e | t <sub>RESW2</sub> | 20                  | _    | μS   | _                |             |
| RES noise canceling width                      |                    | t <sub>RESNCW</sub> | 0.2  | 1.3  | μS               | _           |
| RES setup time                                 |                    | t <sub>RESS</sub>   | 1.3  | _    | μS               | _           |
| NMI setup time                                 |                    | t <sub>NMIS</sub>   | 330  | _    | ns               | Figure 32.7 |
| NMI hold time                                  |                    | t <sub>nmih</sub>   | 330  | _    | ns               |             |
| IRQ7 to IRQ0 setup time                        |                    | t <sub>IRQS</sub>   | 24   | _    | ns               |             |
| IRQ7 to IRQ0 hold time (edge d                 | letection)         | t <sub>IRQH</sub>   | 24   | _    | ns               |             |

Note: \* Wait time to discharge internal high voltage for flash programming/erasing.

# [Operating precautions]

- 1. Mode setup time during power-on reset by the RES pin depends on the combination of signals to be input to the FWE, MD4 to MD0, MD\_CLK1 to MD\_CLK0, and MD\_CLKP pins. If a low-level signal is input to the RES pin while this LSI operates by inputting a mode specified in table 32.3 to the FWE, MD4 to MD0, MD\_CLK1 to MD\_CLK0, and MD\_CLK pins, the mode setup time is defined by t<sub>MDS2</sub>. If a signal other than the combination of signals specified in table 32.3 (undefined mode) is input to the FWE, MD4 to MD0, MD\_CLK1, MD\_CLK0, and MD\_CLKP pins, the mode setup time is defined by t<sub>MDS1</sub>.
- 2. The RES, NMI, and IRQ7–IRQ0 signals are asynchronous inputs, but when the setup times shown here are provided, the signals are considered to have been changed at clock rise. If the setup times are not provided, recognition is delayed until the next clock rise.

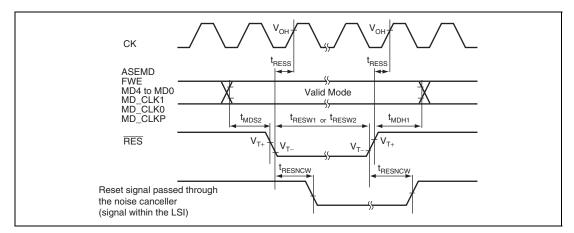


Figure 32.6 Reset Input Timing

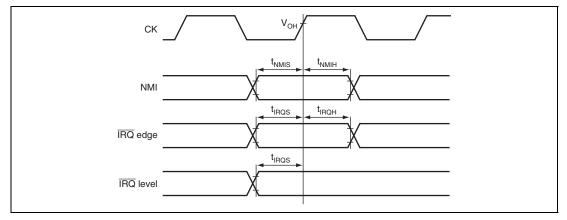


Figure 32.7 Interrupt Signal Input Timing

## 32.3.5 Bus Timing

Table 32.17 shows the bus timing.

## **Table 32.17 Bus Timing**

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AVREFH_A = AVREFH_B = 4.5 \text{ V} \text{ to } AV_{cc}, V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 \text{ V} When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}, V_{cc} = PV_{cc}1. T_a = -40^{\circ}\text{C} \text{ to } 125^{\circ}\text{C}$$$

| Item                      | Symbol            | Min.                     | Max.                     | Unit | Figures               |
|---------------------------|-------------------|--------------------------|--------------------------|------|-----------------------|
| Address delay time        | t <sub>AD</sub>   | _                        | 35                       | ns   | Figures 32.8 to 32.13 |
| Address setup time        | t <sub>AS</sub>   | 0                        | _                        | ns   | Figures 32.8 to       |
| Address hold time         | t <sub>AH</sub>   | 0                        | _                        | ns   | 32.11                 |
| CS delay time             | t <sub>csd</sub>  | _                        | 30                       | ns   | Figures 32.8 to       |
| Read/write delay time     | t <sub>rwd</sub>  | _                        | 30                       | ns   | 32.13                 |
| Read strobe delay time    | t <sub>rsd</sub>  | 1/2t <sub>cyc</sub>      | 1/2t <sub>cyc</sub> + 30 | ns   |                       |
| Read data setup time      | t <sub>RDS</sub>  | 1/2t <sub>cyc</sub> + 15 | _                        | ns   |                       |
| Read data hold time       | t <sub>rdh</sub>  | 0                        | _                        | ns   |                       |
| Write enable delay time 1 | t <sub>weD1</sub> | 1/2t <sub>cyc</sub>      | 1/2t <sub>cyc</sub> + 30 | ns   | Figures 32.8 to 32.12 |
| Write enable delay time 2 | t <sub>wed2</sub> | _                        | 30                       | ns   | Figure 32.13          |
| Write data delay time     | t <sub>wdd</sub>  | _                        | 30                       | ns   | Figures 32.8 to       |
| Write data hold time      | t <sub>wdh</sub>  | 0                        | _                        | ns   | 32.13                 |
| WAIT setup time           | t <sub>wts</sub>  | 1/2t <sub>cyc</sub> + 15 | _                        | ns   | Figures 32.9 to       |
| WAIT hold time            | t <sub>wth</sub>  | 1/2t <sub>cyc</sub> + 0  | _                        | ns   | 32.13                 |

# [Operating precautions]

The guaranteed operating range of power supply  $PV_{cc}1$  in the MCU expanded modes is only  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ . Do not use a voltage outside this range.

Note: 1/2t<sub>cyc</sub> for the delay, setup, and hold time in the above table indicates the falling edge of the clock signal.

Page 1741 of 1824

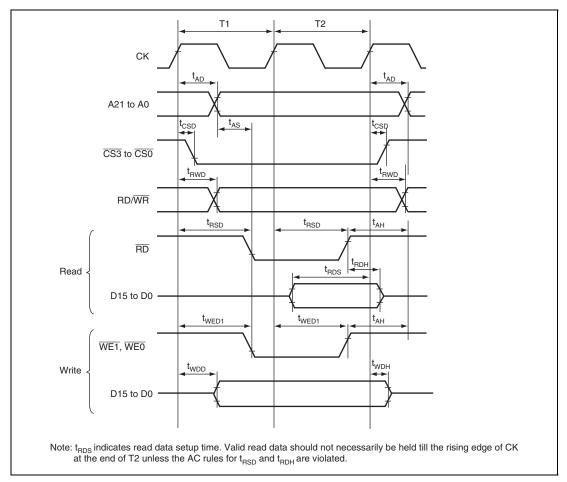


Figure 32.8 Bus Timing (No Waits)

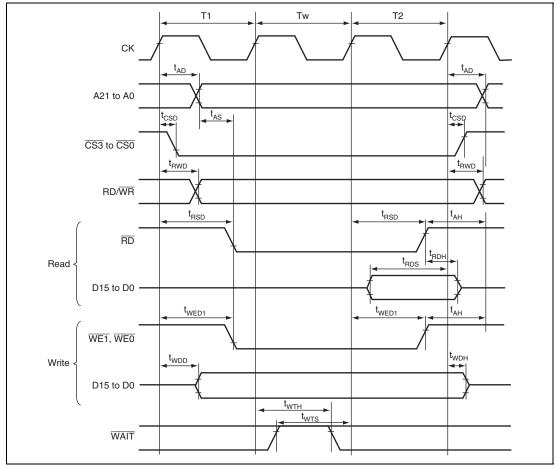


Figure 32.9 Bus Timing (One Software Wait Cycle)

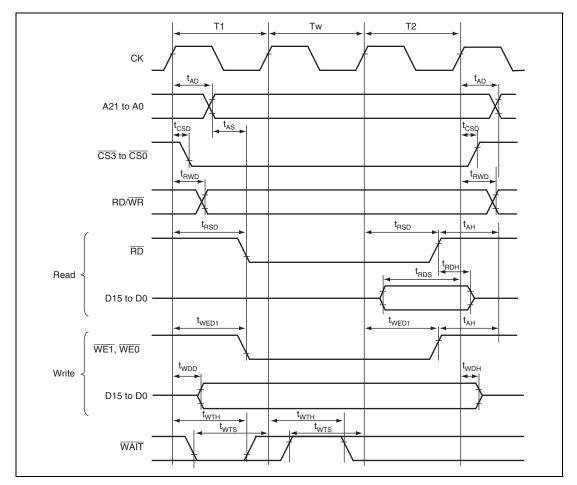


Figure 32.10 Bus Timing (One External Wait Cycle)

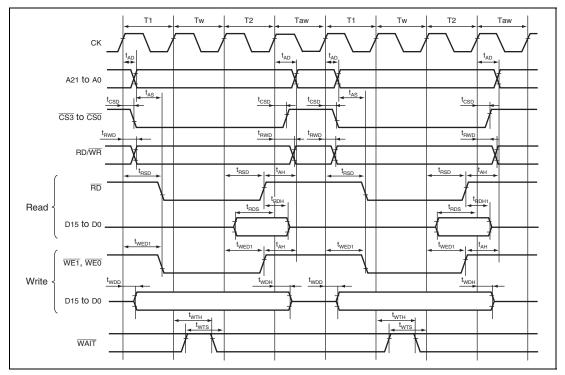


Figure 32.11 Bus Timing

(One Software Wait Cycle, One External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)

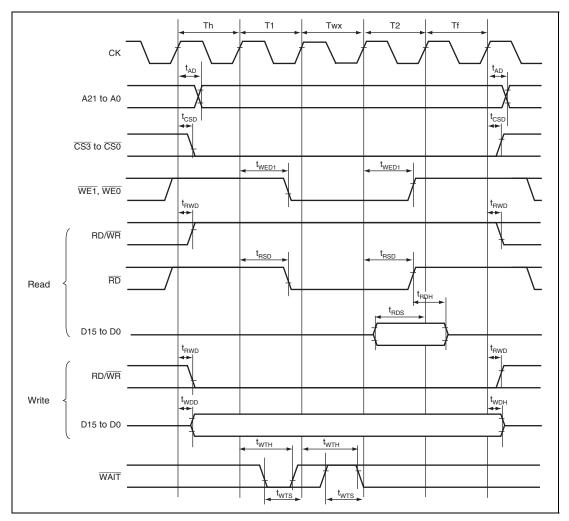


Figure 32.12 Bus Timing of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle UB/LB Control))

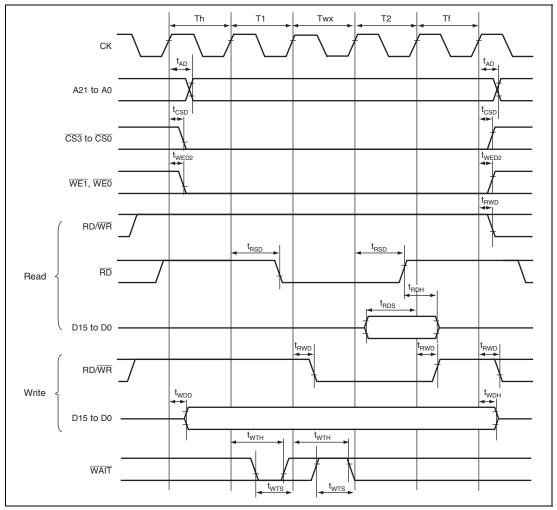


Figure 32.13 Bus Timing of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle WE Control))

#### 32.3.6 Advanced Timer Unit Timing and Advanced Pulse Controller Timing

Table 32.18 shows the advanced timer unit timing and advanced pulse controller timing.

## **Table 32.18 Advanced Timer Unit Timing**

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AVREFH_A = AVREFH_B = 4.5 \text{ V} \text{ to } AV_{cc}, V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 \text{ V} When PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}, V_{cc} = PV_{cc}1. T_a = -40^{\circ}\text{C} \text{ to } 125^{\circ}\text{C}$$

| Item                                   |      | Symbol                                 | Min. | Typ.*1 | Max. | Unit             | Figures      |
|--|------|--|------|--------|------|------------------|--------------|
| Output compare output delay time       |      | t <sub>tocd</sub>                      | _    | _      | 100  | ns               | Figure 32.14 |
| Output compare output rise/fall time*2 | Slow | $t_{_{\rm SR}},t_{_{\rm SF}}$          | _    | 50     | 100  | ns               | Figure 32.29 |
|  | Fast | t <sub>fr</sub> , t <sub>ff</sub>      | _    | 6      | 12   | =                |              |
| Input capture input setup time         |      | t <sub>rics</sub>                      | 24   | _      | _    | ns               | Figure 32.14 |
| Timer clock input setup time           |      | t <sub>TCKS</sub>                      | 24   | _      | _    | ns               | Figure 32.15 |
| Timer clock pulse width (single edge   |      | t <sub>тскwн</sub>                     | 1.5  | _      | _    | t <sub>cyc</sub> | _            |
| specified)                             |      | $\mathbf{t}_{\scriptscriptstyleTCKWL}$ |      |        |      |                  | _            |
| Timer clock pulse width (both edges    |      | t <sub>TCKWH</sub>                     | 2.5  | _      | _    | t <sub>cyc</sub> |              |
| specified)                             |      | t <sub>TCKWL</sub>                     |      |        |      |                  |              |

Notes: 1. The Typ condition is  $V_{cc} = 3.3V$ ,  $PV_{cc}1 = PV_{cc}2 = 5.0V$ ,  $T_a = 25$ °C.

2. Refer to table 32.2 for the pins that can be switched to the faster slew rate.

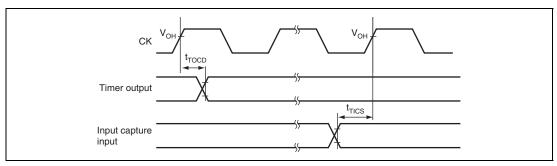


Figure 32.14 ATU Input/Output Timing

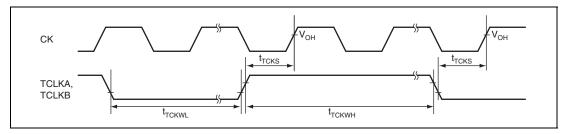


Figure 32.15 ATU Clock Input Timing

## 32.3.7 I/O Port Timing

Table 32.19 shows the I/O port timing.

## Table 32.19 I/O Port Timing

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$$

$$AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$$
,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

$$V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$$

When 
$$PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$$
,  $V_{cc} = PV_{cc}1$ .

 $T_a = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item   |      | Symbol                                  | Min.                  | Typ.* <sup>1</sup> | Max. | Unit | Figures      |
|--|------|---|-----------------------|--------------------|------|------|--------------|
| Port output data delay time                          |      | t <sub>PWD</sub>                        | _                     | _                  | 100  | ns   | Figure 32.16 |
| Port output data rise/fall time*2                    | Slow | $t_{_{\rm SR}},t_{_{\rm SF}}$           | _                     | 50                 | 100  | ns   | Figure 32.29 |
|  | Fast | $t_{_{\mathrm{FR}}},t_{_{\mathrm{FF}}}$ | _                     | 6                  | 12   |      |              |
| Port input setup time                                |      | t <sub>PRS</sub>                        | 24                    | _                  | _    | ns   | Figure 32.16 |
| Port input hold time (level-input except for port G) |      | t <sub>PRH</sub>                        | 24                    | _                  | _    | ns   | _            |
| Port input hold time (port G edge-input)             |      | t <sub>PRH</sub>                        | t <sub>cyc</sub> + 24 | _                  | _    | ns   | _            |

Notes: 1. The Typ condition is  $V_{cc} = 3.3V$ ,  $PV_{cc}1 = PV_{cc}2 = 5.0V$ ,  $T_a = 25$ °C.

2. Refer to table 32.2 for the pins that can be switched to the faster slew rate.

# [Operating precautions]

The guaranteed operating range of power supply  $PV_{cc}1$  in MCU single-chip mode is only  $PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}$ . Do not use a voltage outside this range.

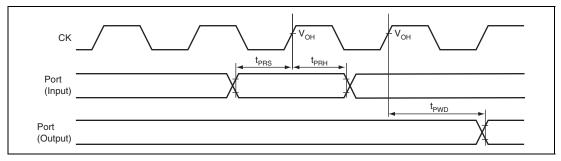


Figure 32.16 I/O Port Input/Output Timing

## 32.3.8 Watchdog Timer Timing

Table 32.20 shows the watchdog timer timing.

#### **Table 32.20 Watchdog Timer Timing**

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}1 = 4.5 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 4.5 \text{ V}, AV$$

| Item              | Symbol            | Min. | Max. | Unit | Figures      |
|-------------------|-------------------|------|------|------|--------------|
| WDTOVF delay time | t <sub>wovd</sub> | _    | 100  | ns   | Figure 32.17 |

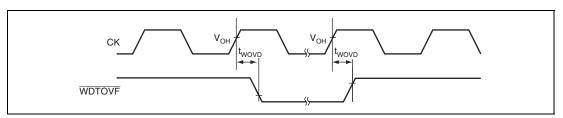


Figure 32.17 Watchdog Timer Timing

## 32.3.9 Serial Communications Interface Timing

Table 32.21 shows the serial communications interface timing.

## **Table 32.21 Serial Communications Interface Timing**

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_{a} = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item  |                 | Symbol            | Min.                          | Typ.* | Max.                          | Unit              | Figures      |
|---|-----------------|-------------------|-------------------------------|-------|-------------------------------|-------------------|--------------|
| Clock cycle (clock sync: input)                         | MD_CLKP = 0     | t <sub>scyc</sub> | 12                            | _     | _                             | t <sub>cyc</sub>  | Figure 32.18 |
| Clock cycle (clock sync: output)                        | _               | t <sub>scyc</sub> | 8                             | _     | _                             | t <sub>cyc</sub>  | _            |
| Clock cycle (clock sync: input)                         | MD_CLKP = 1     | t <sub>scyc</sub> | 16                            | _     | _                             | t <sub>cyc</sub>  | -            |
| Clock cycle (clock sync: output)                        | _               | t <sub>scyc</sub> | 16                            | _     | _                             | t <sub>cyc</sub>  | -            |
| Clock pulse width                                       |                 | t <sub>sckw</sub> | 0.4                           | _     | 0.6                           | t <sub>scyc</sub> | _            |
| Input clock rise time                                   | MD_CLKP = 0     | t <sub>sckr</sub> |                               | _     | 0.8                           | t <sub>cyc</sub>  | _            |
| Input clock fall time                                   | _               | t <sub>sckf</sub> | _                             | _     | 0.8                           | t <sub>cyc</sub>  | _            |
| Input clock rise time                                   | MD_CLKP = 1     | t <sub>sckr</sub> | _                             | _     | 1.6                           | t <sub>cyc</sub>  | _            |
| Input clock fall time                                   | _               | t <sub>sckf</sub> | _                             | _     | 1.6                           | t <sub>cyc</sub>  | _            |
| Synchronous mode transmit data delay time               | MD_CLKP = 0     | t <sub>TxD</sub>  | _                             | _     | 3 × t <sub>cyc</sub><br>+ 100 | ns                | Figure 32.19 |
| (SCK is input)  | MD_CLKP = 1     | _                 |                               | _     | 3 × t <sub>cyc</sub> + 75     | _                 |              |
| Synchronous mode receive data setup time (SCK is input) |                 | t <sub>RxS</sub>  | -1 × t <sub>cyc</sub><br>+ 30 | _     | _                             | ns                | -            |
| Synchronous mode receive data hold time (SCK is input)  |                 | t <sub>RxH</sub>  | 2 × t <sub>cyc</sub> + 30     | _     | _                             | ns                | _            |
| Synchronous mode transtime (SCK is output)              | smit data delay | t <sub>TxD</sub>  | _                             | _     | 70                            | ns                |              |

| Item   |   |         | Symbol                                      | Min.                       | Тур.* | Max. | Unit | Figures      |
|--|---|---------|---|----------------------------|-------|------|------|--------------|
| Synchronous mode   | Transmission                            | Slow    | t <sub>RxS</sub>                            | 120                        | _     | _    | ns   | Figure 32.19 |
| receive data setup and time (SCK is output) reception, transmission only |   | Fast    |   | 30                         | _     | _    |      |              |
|  | Reception                               | Slow    | _   | 120 +                      |       | _    | ="   |              |
|  | only                                    |         | _   | $3 \times t_{\text{cyc}}$  |       |      | _    |              |
|  |   | Fast    |   | 30 + 3<br>$\times t_{cyc}$ | —     | —    |      |              |
| -  |   |         |   |                            |       |      |      | _            |
| Synchronous mode re (SCK is output)                                      | eceive data hol                         | d time  | t <sub>RxH</sub>                            | 1 × t <sub>cyc</sub> + 30  | _     | _    | ns   |              |
| Asynchronous mode time   | transmit data d                         | elay    | t <sub>TxD</sub>                            | _                          | _     | 100  | ns   | -            |
| Asynchronous mode time   | receive data se                         | tup     | t <sub>RxS</sub>                            | 100                        | _     | _    | ns   | _            |
| Asynchronous mode  | receive data ho                         | ld time | t <sub>RxH</sub>                            | 100                        | _     | _    | ns   | -            |
| SCK clock output rise  | /fall time                              | Slow    | $t_{_{\rm SR}},t_{_{\rm SF}}$               | _                          | 50    | 100  | ns   | Figure 32.29 |
|  |   | Fast    | $t_{_{\mathrm{FR}}}$ , $t_{_{\mathrm{FF}}}$ | _                          | 6     | 12   | =    |              |
|  | TxD Transmit data output rise/fall Slow |         | $t_{_{\rm SR}},t_{_{\rm SF}}$               |                            | 50    | 100  | ns   | -            |
| time   |   | Fast    | $t_{_{\mathrm{FR}}},t_{_{\mathrm{FF}}}$     | _                          | 6     | 12   |      |              |

Note: \* The Typ condition is  $V_{cc} = 3.3V$ ,  $PV_{cc}1 = PV_{cc}2 = 5.0V$ ,  $T_a = 25^{\circ}C$ .

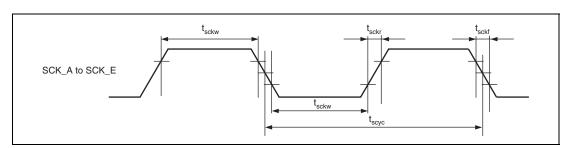


Figure 32.18 SCK Input Timing

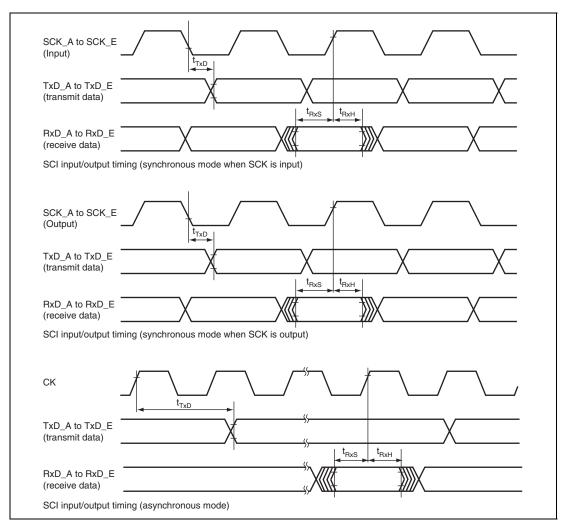


Figure 32.19 SCI Input/Output Timing

## **32.3.10** CAN Timing

Table 32.22 shows the CAN timing.

## Table 32.22 CAN Timing

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}3 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}4 = 4.5 \text{ V} \pm 0.5 \text$$

| Item                         |             | Symbol                            | Min. | Typ.*1 | Max. | Unit | Figures        |
|------------------------------|-------------|-----------------------------------|------|--------|------|------|----------------|
| CAN Transmit data delay time |             | t <sub>CTxD</sub>                 | _    | _      | 100  | ns   | Figure 32.20   |
| CTx Transmit data output     | Slow        | t <sub>sr</sub> , t <sub>sf</sub> | _    | 50     | 100  | ns   | Figure 32.29   |
| rise/fall time               | Fast        | $t_{FR}, t_{FF}$                  | _    | 6      | 12   | _    |                |
| CAN Receive data setup time  |             | t <sub>CRxS</sub>                 | 100  | _      | _    | ns   | Figure 32.20   |
| CAN Receive data hold time   |             | t <sub>CRxH</sub>                 | 100  | _      | _    | ns   | -              |
| CAN bus jitter*3             | Period 1µs  | f <sub>CANJIT</sub>               |      | 0.13*2 | _    | %    | 1Mbps/1-bit    |
|                              | Period 2μs  | _                                 | _    | 0.08*2 | 0.1  | %    | 500kbps/1-bit  |
|                              | Period 10μs | _                                 | _    | 0.02*2 | _    | %    | 1Mbps/10-bit   |
|                              | Period 20μs | _                                 | _    | 0.01*2 | 0.1  | %    | 500kbps/10-bit |

Notes: 1. The Typ condition is  $V_{cc} = 3.3V$ ,  $PV_{cc}1 = PV_{cc}2 = 5.0V$ ,  $T_a = 25^{\circ}C$ .

- 2. The typ value of CAN bus jitter is measured value of the representative sample without crystal deviation.
  - It is not a guarantee value to which all samples are tested.
- 3. The crystal deviation is not included.

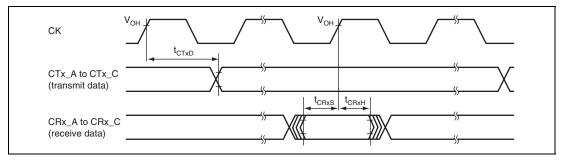


Figure 32.20 CAN Input/Output Timing

## **32.3.11** SPI Timing

Tables 32.23 (1) and 32.23 (2) show the SPI timing.

## Table 32.23 (1) SPI Timing (when Output Slew Rates are Fast)

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL\_A = AVREFL\_B = 0 V$ 

When  $PV_{CC}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC}1$ .

 $T_{a} = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item                               |                   |        | Symbol               | Min.   | Тур.* | Max. | Unit             | Figures      |
|------------------------------------|-------------------|--------|----------------------|--|-------|------|------------------|--------------|
| RSPCK                              | MD_CLKP = 0       | Master | t <sub>SPcyc</sub>   | 2  | _     | 4096 | t <sub>cyc</sub> | Figure 32.21 |
| clock cycle                        |                   | Slave  | _                    | 8  | _     | 4096 | _                |              |
|                                    | MD_CLKP = 1       | Master | _                    | 4  | _     | 4096 | t <sub>cyc</sub> | _            |
|                                    |                   | Slave  | _                    | 16   | _     | 4096 | _                |              |
| RSPCK clock high-level pulse width |                   | Master | t <sub>spckwh</sub>  | (t <sub>SPCVC</sub> - t <sub>SPCKR</sub> -       | _     |      | ns               | -            |
|                                    |                   |        | _                    | t <sub>SPCKF</sub> )<br>/2 - 5                   |       |      |                  | _            |
|                                    |                   | Slave  |                      | $(t_{_{\mathrm{SPCKR}}} - t_{_{\mathrm{SPCKR}}}$ | _     | _    | ns               |              |
|                                    |                   |        |                      | t <sub>SPCKF</sub> )<br>/2                       |       |      |                  | _            |
| RSPCK cloc<br>pulse width          | ck low-level      | Master | t <sub>SPCKWL</sub>  | $(t_{\text{SPCKR}} - t_{\text{SPCKR}} -$         | _     | _    | ns               |              |
|                                    |                   |        |                      | $t_{SPCKF}$ ) $/2-5$                             |       |      |                  |              |
|                                    |                   | Slave  | _                    | (t <sub>SPCYC</sub> - t <sub>SPCKR</sub> -       | _     | _    | ns               | -            |
|                                    |                   |        |                      | $t_{\text{SPCKF}}$ ) /2                          |       |      |                  |              |
| RSPCK cloc                         | ck rise/fall time | Output | t <sub>spckR</sub> , |  | 3     | 5    | ns               | _            |
|                                    |                   | Input  | t <sub>SPCKF</sub>   | _  |       | 1    | μS               |              |

| Item                        |        | Symbol             | Min.                      | Тур.* | Max.                    | Unit                        | Figures       |
|-----------------------------|--------|--------------------|---------------------------|-------|-------------------------|-----------------------------|---------------|
| Data input setup time       | Master | t <sub>su</sub>    | 18                        | _     | _                       | ns                          | Figures 32.22 |
|                             | Slave  | <del>_</del>       | 20 – 2                    | _     | _                       | ns                          | to 32.25      |
|                             |        |                    | $\times$ t <sub>cyc</sub> |       |                         |                             | _             |
| Data input hold time        | Master | t <sub>H</sub>     | 0                         | _     | _                       | ns                          |               |
|                             | Slave  |                    | 20 + 2                    | —     | —                       |                             |               |
|                             |        |                    | $\times$ t <sub>cyc</sub> |       |                         |                             | _             |
| SSL setup time              | Master | t <sub>LEAD</sub>  | 1                         | _     | 8                       | t <sub>SPcyc</sub>          | _             |
|                             | Slave  |                    | 4                         | _     | _                       | $t_{\scriptscriptstylecyc}$ | _             |
| SSL hold time               | Master | t <sub>LAG</sub>   | 1                         | _     | 8                       | t <sub>SPcyc</sub>          | _             |
|                             | Slave  | _                  | 4                         | _     | _                       | t <sub>cyc</sub>            | _             |
| Data output delay time      | Master | t <sub>od</sub>    | _                         | _     | 15                      | ns                          | _             |
|                             | Slave  | _                  | _                         | _     | 25 + 3                  | _                           |               |
|                             |        |                    |                           |       | $\times$ t $_{\rm cyc}$ |                             | _             |
| Data output hold time       | Master | t <sub>oн</sub>    | 0                         | _     | _                       | ns                          | _             |
|                             | Slave  | _                  | 0                         | _     | _                       | _                           |               |
| Continuous transmission     | Master | t <sub>TD</sub>    | t <sub>SPcyc</sub> +      | _     | 8                       | ns                          | _             |
| delay time                  |        |                    | $2 \times t_{\text{cyc}}$ |       | × t <sub>SPcyc</sub>    |                             |               |
|                             |        |                    |                           |       | + 2 × t <sub>cyc</sub>  |                             |               |
|                             | Slave  | _                  | $4 \times t_{\text{cyc}}$ | _     | -cyc                    |                             | _             |
| MOSI, MISO rise/fall time   | Output | t                  | - Cyc                     | 3     | 5                       | ns                          | _             |
| WOOI, WIOO 1130/1dil tillic | Input  | t                  |                           |       | 1                       |                             | _             |
| 001 /-                      | -      | t <sub>DF</sub>    |                           |       |                         | μS                          | _             |
| SSL rise/fall time          | Output | t <sub>sslr,</sub> |                           | 3     | 5                       | ns                          | _             |
|                             | Input  | t <sub>sslf</sub>  | _                         |       | 1                       | μS                          |               |
| Slave access time           |        | t <sub>sa</sub>    |                           | _     | 4                       | t <sub>cyc</sub>            | Figures 32.24 |
| Slave out release time      |        | $t_{_{REL}}$       |                           | _     | 3                       | t <sub>cyc</sub>            | and 32.25     |

Note: \* The Typ condition is  $V_{cc} = 3.3V$ ,  $PV_{cc}1 = PV_{cc}2 = 5.0V$ ,  $T_a = 25^{\circ}C$ .

Timing on condition that the slew rates of RSPCK, MISO/MOSI, and SSL are all fast.

## Table 32.23 (2) SPI Timing (when Output Slew Rates are Slow)

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_{a} = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item                    |                |        | Symbol               | Min.   | Тур.* | Max. | Unit             | Figures      |
|-------------------------|----------------|--------|----------------------|--|-------|------|------------------|--------------|
|                         | MD_CLKP = 0    | Master | t <sub>SPcyc</sub>   | 8  | _     | 4096 | t <sub>cyc</sub> | Figure 32.21 |
| cycle                   |                | Slave  | _                    | 8  | _     | 4096 | _                |              |
|                         | MD_CLKP = 1    | Master | _                    | 16   | _     | 4096 | t <sub>cyc</sub> | _            |
|                         |                | Slave  | _                    | 16   | _     | 4096 |                  | _            |
| RSPCK clock pulse width | high-level     | Master | t <sub>SPCKWH</sub>  | ${\rm (t_{\scriptscriptstyle SPCYC}} - {\rm t_{\scriptscriptstyle SPCKR}} -$ | _     | _    | ns               |              |
|                         |                |        | _                    | t <sub>SPCKF</sub> )<br>/2 - 5   |       |      |                  | _            |
|                         |                | Slave  |                      | $(t_{_{\mathrm{SPCKR}}} - t_{_{\mathrm{SPCKR}}}$                             | _     |      |                  |              |
|                         |                |        |                      | t <sub>SPCKF</sub> )<br>/2   |       |      |                  | _            |
| RSPCK clock pulse width | low-level      | Master | t <sub>SPCKWL</sub>  | $(t_{_{\mathrm{SPCKR}}} - t_{_{\mathrm{SPCKR}}}$                             | _     | _    | ns               |              |
|                         |                |        |                      | t <sub>SPCKF</sub> )<br>/2 - 5   |       |      |                  |              |
|                         |                | Slave  | _                    | (t <sub>SPCKR</sub> -  | _     | _    |                  | -            |
|                         |                |        |                      | $t_{\text{SPCKF}}$ ) /2  |       |      |                  |              |
| RSPCK clock             | rise/fall time | Output | t <sub>spckR</sub> , |  | 20    | 40   | ns               | _            |
|                         |                | Input  | t <sub>SPCKF</sub>   | _  | _     | 1    | μS               |              |

| Item                               |        | Symbol             | Min.   | Тур.* | Max.  | Unit               | Figures                   |
|------------------------------------|--------|--------------------|--|-------|---|--------------------|---------------------------|
| Data input setup time              | Master | t <sub>su</sub>    | 100  |       |   | ns                 | Figures 32.22<br>to 32.25 |
|                                    | Slave  |                    | 20 - 2<br>× t <sub>cyc</sub>                 | _     | _   | ns                 | _                         |
| Data input hold time               | Master | t <sub>H</sub>     | 0  | _     | _   | ns                 | _                         |
|                                    | Slave  | _                  | 20 + 2<br>× t <sub>cyc</sub>                 | _     | _   | _                  | _                         |
| SSL setup time                     | Master | t <sub>LEAD</sub>  | 1  |       | 8   | t <sub>SPcyc</sub> | _                         |
|                                    | Slave  |                    | 4  | _     | _   | t <sub>cyc</sub>   | _                         |
| SSL hold time                      | Master | t <sub>LAG</sub>   | 1  | —     | 8   | t <sub>SPcyc</sub> | _                         |
|                                    | Slave  |                    | 4  | _     | _   | t <sub>cyc</sub>   | _                         |
| Data output delay time             | Master | t <sub>od</sub>    |  | _     | 20  | ns                 |                           |
|                                    | Slave  |                    | _  | _     | $\begin{array}{c} 100 \ + \\ 3 \times t_{\text{cyc}} \end{array}$ |                    |                           |
| Data output hold time              | Master | t <sub>oh</sub>    | -10  | _     | _   | ns                 | _                         |
|                                    | Slave  |                    | 0  |       | _   | _'                 | _                         |
| Continuous transmission delay time | Master | t <sub>TD</sub>    | $t_{\text{SPcyc}} + 2 \times t_{\text{cyc}}$ | _     | $8 \\ \times t_{\text{SPcyc}} \\ + 2 \times \\ t_{\text{cyc}}$    | ns                 |                           |
|                                    | Slave  |                    | $4 \times t_{\text{cyc}}$                    | _     | _   |                    | _                         |
| MOSI, MISO rise/fall time          | Output | t <sub>DR,</sub>   | _  | 20    | 40  | ns                 | _                         |
|                                    | Input  | t <sub>DF</sub>    | _  | _     | 1   | μS                 | _                         |
| SSL rise/fall time                 | Output | t <sub>sslr,</sub> |  | 20    | 40  | ns                 | _                         |
|                                    | Input  | t <sub>sslf</sub>  |  |       | 1   | μS                 |                           |
| Slave access time                  |        | t <sub>sa</sub>    |  |       | 4   | t <sub>cyc</sub>   | Figures 32.24             |
| Slave out release time             |        | t <sub>rel</sub>   | —<br>—                                       | _     | 3   | t <sub>cyc</sub>   | and 32.25                 |

Note: \* The Typ condition is  $V_{cc} = 3.3V$ ,  $PV_{cc}1 = PV_{cc}2 = 5.0V$ ,  $T_a = 25^{\circ}C$ . Timing on condition that the slew rates of RSPCK, MISO/MOSI, and SSL are all slow.

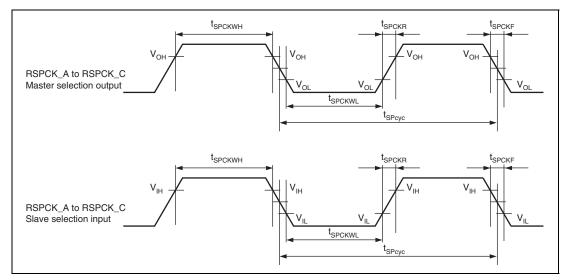


Figure 32.21 SPI Clock Timing

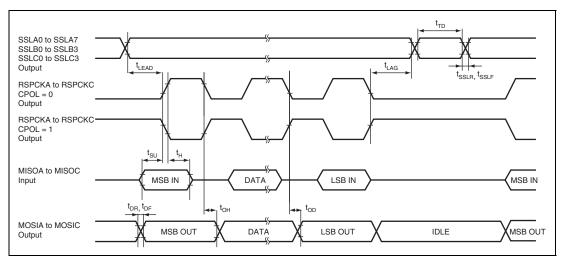


Figure 32.22 SPI Timing (Master, CPHA = 0)

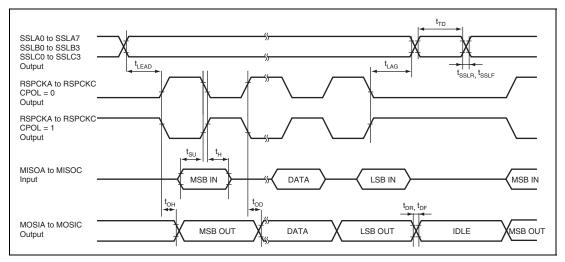


Figure 32.23 SPI Timing (Master, CPHA = 1)

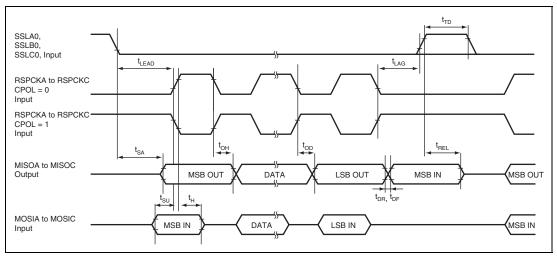


Figure 32.24 SPI Timing (Slave, CPHA = 0)

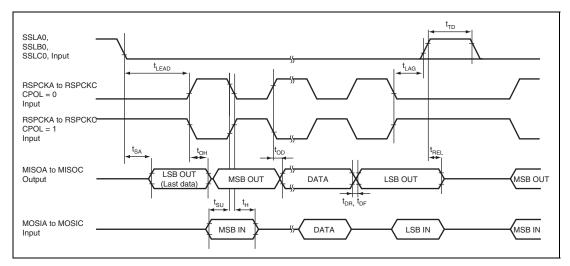


Figure 32.25 SPI Timing (Slave, CPHA = 1)

## 32.3.12 A/D Converter Timing

Table 32.24 shows the A/D converter timing.

#### Table 32.24 A/D Converter Timing

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AVREFH_A = AVREFH_B = 4.5 \text{ V} \text{ to } AV_{cc}, V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 \text{ V} When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}, V_{cc} = PV_{cc}1. T_a = -40^{\circ}\text{C} \text{ to } 125^{\circ}\text{C}$$$

| Item                                    |      | Symbol                            | Min. | Тур.* | Max. | Unit | Figure       |
|---|------|-----------------------------------|------|-------|------|------|--------------|
| External trigger input start delay time |      | t <sub>TRGS</sub>                 | 50   | _     | _    | ns   | Figure 32.26 |
| ADEND output delay time                 |      | t <sub>ADENDD</sub>               | _    | _     | 100  | ns   | Figure 32.27 |
| ADEND output rise/fall time             | Slow | $t_{\rm SR},t_{\rm SF}$           |      | _     | 100  | ns   | Figure 32.29 |
| F                                       | Fast | t <sub>fr</sub> , t <sub>ff</sub> | _    |       | 12   | _    |              |

Note: \* The Typ condition is  $V_{cc} = 3.3V$ ,  $PV_{cc}1 = PV_{cc}2 = 5.0V$ ,  $T_a = 25$ °C.

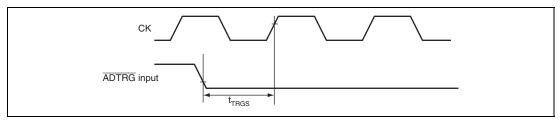


Figure 32.26 External Trigger Input Timing

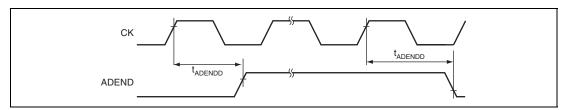


Figure 32.27 Analog Conversion Timing

## 32.3.13 UBC Trigger Timing

Table 32.25 shows the UBC trigger timing.

#### **Table 32.25 UBC Trigger Timing**

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{CC}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC}1$ .

 $T_a = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item                           |      | Symbol  | Min. | Тур.* | Max. | Unit | Figures      |
|--------------------------------|------|---|------|-------|------|------|--------------|
| UBCTRG delay time              |      | t <sub>ubctgd</sub>                                       | _    | _     | 100  | ns   | Figure 32.28 |
| UBCTRG output rise/fall time S | Slow | $\mathbf{t}_{_{\mathrm{SR}}},\mathbf{t}_{_{\mathrm{SF}}}$ | _    | 50    | 100  | ns   | Figure 32.29 |
| F                              | ast  | t <sub>fr</sub> , t <sub>ff</sub>                         | _    | 6     | 12   | _    |              |

Note: \* The Typ condition is  $V_{cc} = 3.3V$ ,  $PV_{cc}1 = PV_{cc}2 = 5.0V$ ,  $T_a = 25$ °C.

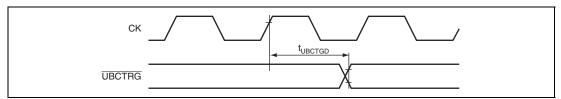


Figure 32.28 UBC Trigger Timing

## 32.3.14 Output Slew Rate

Table 32.26 shows the output slew rate timing.

#### **Table 32.26 Output Slew Rate Timing**

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}1 = 4.5 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{cc}2 = 4.5 \text{ V} \pm 0.5 \text$$

| Item                                      | Symbol                        | Conditions   | Min. | Тур. | Max. | Unit | Figures      |
|---|-------------------------------|--------------|------|------|------|------|--------------|
| Output rise/fall time                     | $t_{_{\rm SR}},t_{_{\rm SF}}$ | Load 25[pF]  | _    | 25   | 50   | ns   | Figure 32.29 |
| Slow slew rate                            |                               | Load 50[pF]  | _    | 50   | 100  | _    |              |
|   |                               | Load 75[pF]  | _    | 75   | 150  | _    |              |
|   |                               | Load 100[pF] | _    | 100  | 200  | _    |              |
| Output rise/fall time $t_{FR}$ , $t_{FF}$ |                               | Load 25[pF]  | _    | 4    | 6    | ns   | <del>-</del> |
| Fast slew rate                            |                               | Load 50[pF]  | _    | 6    | 12   | _    |              |
|   |                               | Load 75[pF]  | _    | 8    | 18   |      |              |
|   |                               | Load 100[pF] | _    | 10   | 24   | _    |              |

Notes: 1. The Typ condition is  $V_{cc} = 3.3V$ ,  $PV_{cc}1 = PV_{cc}2 = 5.0V$ ,  $T_a = 25$ °C.

2. Refer to table 32.2 for the pins that can be switched to the faster slew rate.

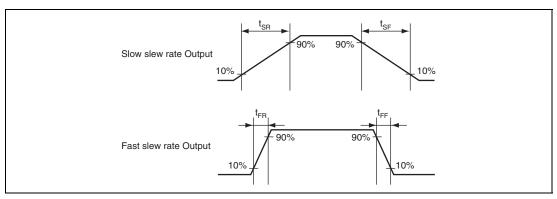


Figure 32.29 Output Slew Rate Timing

# 32.3.15 JTAG Interface Timing

Table 32.27 shows the JTAG interface timing.

#### **Table 32.27 JTAG Interface Timing**

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_a = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item              |             | Symbol              | Min. | Max. | Unit                | Figures      |
|-------------------|-------------|---------------------|------|------|---------------------|--------------|
| TCK clock cycle   | MD_CLKP = 0 | t <sub>TCKeye</sub> | 10   | _    | t <sub>cyc</sub>    | Figure 32.30 |
|                   | MD_CLKP = 1 | <del>_</del>        | 20   | _    |                     |              |
| TCK clock high-le | vel width   | t <sub>тскн</sub>   | 0.4  | 0.6  | t <sub>TCKcyc</sub> | _            |
| TCK clock low-lev | el width    | t <sub>TCKL</sub>   | 0.4  | 0.6  | t <sub>TCKeye</sub> | _            |
| TRST pulse width  |             | t <sub>rrsw</sub>   | 20   | _    | t <sub>TCKcyc</sub> | Figure 32.31 |
| TMS setup time    |             | t <sub>mss</sub>    | 30   | _    | ns                  | Figure 32.32 |
| TMS hold time     |             | t <sub>msh</sub>    | 30   | _    | ns                  | _            |
| TDI setup time    |             | t <sub>TDIS</sub>   | 30   | _    | ns                  | _            |
| TDI hold time     |             | t <sub>tdih</sub>   | 30   | _    | ns                  |              |
| TDO delay time    |             | t <sub>tdod</sub>   | _    | 45   | ns                  |              |

# [Operating precautions]

The JTAG Interface pins constitute a circuit requiring the voltage of  $V_{cc}$  = 3.3 V ±0.3 V. Comply with the input and output voltages specified in the DC characteristics, for operation.

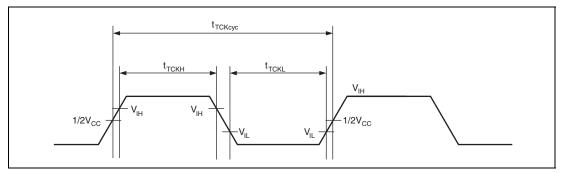


Figure 32.30 JTAG Interface Clock Timing

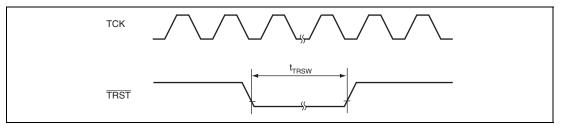


Figure 32.31 JTAG Interface TRST Timing

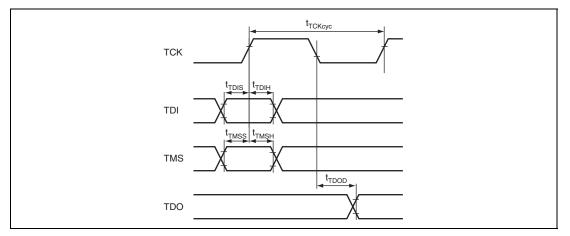


Figure 32.32 JTAG Interface Input/Output

## **32.3.16** AUD Timing

Table 32.28 shows the AUD timing.

## **Table 32.28 AUD Timing**

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{cc} = PV_{cc}1$ .

 $T_{a} = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item                                  |                   | Symbol                | Min. | Max. | Unit                  | Figures      |
|---------------------------------------|-------------------|-----------------------|------|------|-----------------------|--------------|
| AUDRST setup time                     | (trace mode)      | t                     | 30   | _    | ns                    | Figure 32.33 |
| AUDRST pulse                          | MD_CLKP = 0       | t <sub>AURSTTW</sub>  | 10   | _    | t <sub>cyc</sub>      | _            |
| width (trace mode)                    | MD_CLKP = 1       |                       | 20   | _    | <del>_</del>          |              |
| AUDMD setup time                      | MD_CLKP = 0       | t                     | 10   | _    | t <sub>cyc</sub>      | <del></del>  |
| (trace mode)                          | MD_CLKP = 1       | <del></del>           | 20   | _    | _                     |              |
| AUDCK cycle time                      | MD_CLKP = 0       | t <sub>AUCKTcyc</sub> | 1    | 2.5  | t <sub>cyc</sub>      | _            |
| (trace mode)                          | MD_CLKP = 1       | <del></del>           | 2    | 5    | _                     |              |
| AUDCK high-level w                    | idth (trace mode) | t <sub>auckth</sub>   | 0.4  | 0.6  | t <sub>AUCKTeye</sub> | _            |
| AUDCK low-level wid                   | dth (trace mode)  | t <sub>AUCKTL</sub>   | 0.4  | 0.6  | t <sub>AUCKTeye</sub> | _            |
| Trace data output de                  | elay time         | t <sub>AUDTTD</sub>   | _    | 20   | ns                    | Figure 32.34 |
| AUDSYNC output de                     | elay time         | t <sub>AUSYTD</sub>   | _    | 20   | ns                    | _            |
| AUDCK cycle time                      | MD_CLKP = 0       | t <sub>AUCKMcyc</sub> | 1    | _    | t <sub>cyc</sub>      | Figure 32.35 |
| (monitor mode)                        | MD_CLKP = 1       |                       | 2    | _    | <del>_</del>          |              |
| AUDCK high-level width (monitor mode) |                   | t <sub>auckmh</sub>   | 0.4  | _    | t <sub>AUCKMeye</sub> | <del></del>  |
| AUDCK low-level width (monitor mode)  |                   | t <sub>auckml</sub>   | 0.4  | _    | t <sub>AUCKMeye</sub> | _            |
| AUDRST setup time (monitor mode)      |                   | t <sub>AURSTMS</sub>  | 30   | _    | ns                    | _            |
| AUDRST pulse widtl                    | n (monitor mode)  | t <sub>AURSTMW</sub>  | 5    | _    | t <sub>AUCKMeye</sub> | <del></del>  |
| AUDMD setup time (                    | (monitor mode)    | t <sub>AUMDMS</sub>   | 5    | —    | t <sub>AUCKMeye</sub> |              |

| Item                           | Symbol              | Min. | Max. | Unit | Figures      |
|--------------------------------|---------------------|------|------|------|--------------|
| Monitor data output delay time | t <sub>AUDTMD</sub> | _    | 35   | ns   | Figure 32.36 |
| Monitor data input setup time  | t <sub>AUDTMS</sub> | 15   | _    | ns   |              |
| Monitor data input hold time   | t <sub>audtmh</sub> | 5    | _    | ns   |              |
| AUDSYNC input setup time       | t <sub>AUDSYS</sub> | 15   | _    | ns   |              |
| AUDSYNC input hold time        | t <sub>AUDSYH</sub> | 5    | _    | ns   |              |

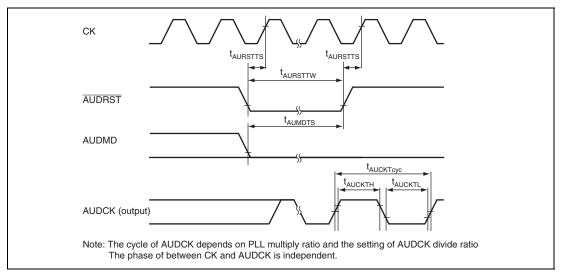


Figure 32.33 Trace Mode Reset Timing

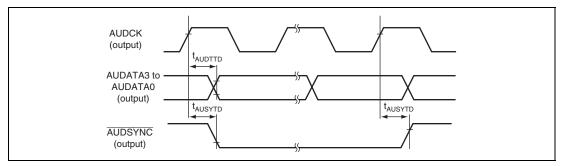


Figure 32.34 Trace Mode Timing

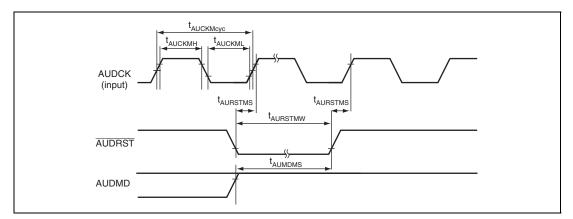


Figure 32.35 Monitor Mode Reset Timing

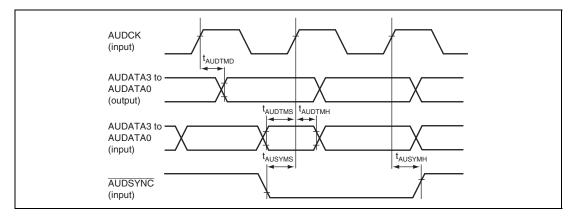


Figure 32.36 Monitor Mode Timing

### 32.3.17 Measuring Conditions for AC Characteristics

Input reference levels High level:  $V_{IH}$  min. value, low level:  $V_{IL}$  max. value

Output reference level High level: 2.0 V, Low level: 0.8 V

Input rise and fall times: 1ns

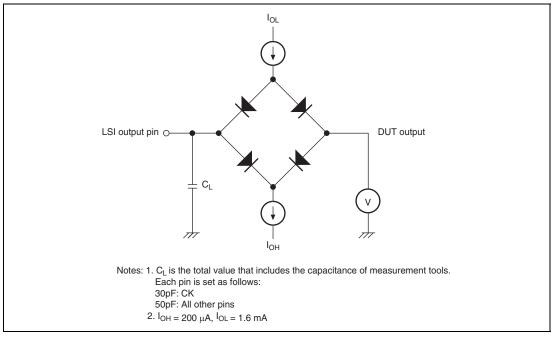


Figure 32.37 Output Test Circuit

#### 32.4 A/D Converter Characteristics

Table 32.29 shows the A/D converter characteristics.

#### Table 32.29 A/D Converter Characteristics

Conditions:  $V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V},$ 

 $PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V},$ 

 $AV_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AVREFH\_A = AVREFH\_B = 4.5 \text{ V}$  to  $AV_{cc}$ ,

 $V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V$ 

When  $PV_{CC}1 = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC}1$ .

 $T_a = -40^{\circ} \text{C} \text{ to } 125^{\circ} \text{C}$ 

| Item   |          | Symbol | Min. | Max. | Unit             | Figures |
|--|----------|--------|------|------|------------------|---------|
| Digit resolution                                 |          | _      | 12   | 12   | bit              |         |
| Voltage resolution*1                             |          | _      | 1.10 | 1.34 | mV               |         |
| A/D conversion cycle*2                           |          | _      | 25   | 50   | t <sub>cyc</sub> |         |
| A/D conversion time $(f_{OP} = 20[MHz]/40[MHz])$ |          | _      | 1.25 | 2.5  | μS               |         |
| Non-linear error                                 |          | _      | _    | ±4.0 | LSB              |         |
| Offset error                                     |          |        | _    | ±7.5 | LSB              |         |
| Full-scale error                                 |          | _      | _    | ±7.5 | LSB              |         |
| Quantization error                               |          | _      | _    | ±0.5 | LSB              |         |
| Absolute error                                   |          |        | _    | ±8.0 | LSB              |         |
| Absolute error at self-diagr                     | nosis    | _      | _    | ±8.0 | LSB              |         |
| Analog input capacitance                         | Awaiting | _      | _    | 20   | pF               |         |
|  | Sampling |        | _    | 40   | pF               |         |
| Permitted analog signal source impedance         |          |        |      | 3    | kΩ               |         |

Notes: 1. At AVREFH-AVREFL = 4.5 V, resolution is 1.10mV. At AVREFH-AVREFL = 5.5 V, resolution is 1.34mV.

2. A/D conversion cycle depends on CKS bit setting in ADCER0/1 register.

## 32.5 Flash Memory Characteristics

Table 32.30 shows the flash memory characteristics. The characteristics are measured under the condition that the frequency of the clock signal input on the EXTAL pin is 20 MHz.

#### **Table 32.30 Flash Memory Characteristics**

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AVREFH_A = AVREFH_B = 4.5 \text{ V} \text{ to } AV_{cc}, V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 \text{ V} When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}, V_{cc} = PV_{cc}1. T_a = -40^{\circ}\text{C} \text{ to } 125^{\circ}\text{C}$$$

| Item             |   |  | Symbol             | Min. | Тур. | Max. | Unit | Figures |
|------------------|---|--|--------------------|------|------|------|------|---------|
| Programming time | 256 bytes   | Programming/erasing: 100 times or less | t <sub>P256</sub>  | _    | 2    | 12   | ms   |         |
|                  |   | Programming/erasing: 101 to 1000 times | _                  | _    | 2.4  | 14.4 | _    |         |
|                  | 8 Kbytes  | Programming/erasing: 100 times or less | t <sub>P8K</sub>   | _    | 45   | 100  | ms   | _       |
|                  |   | Programming/erasing: 101 to 1000 times | _                  | _    | 54   | 120  | _    |         |
|                  |   | Programming/erasing: 100 times or less | t <sub>E8K</sub>   | _    | 50   | 120  | ms   |         |
|                  |   | Programming/erasing: 101 to 1000 times |                    | _    | 60   | 144  | _    | _       |
|                  | 64 Kbytes   | Programming/erasing: 100 times or less | t <sub>E64K</sub>  | _    | 400  | 875  | ms   |         |
|                  |   | Programming/erasing: 101 to 1000 times | _                  | _    | 480  | 1050 | _    |         |
|                  | 128 Kbytes  | Programming/erasing: 100 times or less | t <sub>E128K</sub> | _    | 800  | 1750 | ms   | _       |
|                  |   | Programming/erasing: 101 to 1000 times | _                  | _    | 960  | 2100 | _    |         |
|                  | 32 Kbytes* Programming/erasing: 100 times or less |  | t <sub>E32K</sub>  | _    | 200  | 480  | ms   | _       |
|                  |   | Programming/erasing: 101 to 1000 times | _                  | _    | 240  | 576  | _    |         |

| Item   | Symbol             | Min. | Тур. | Max. | Unit  | Figures            |
|--|--------------------|------|------|------|-------|--------------------|
| Reprogramming/Erasing cycle                                      | N <sub>PEC</sub>   | _    | _    | 1000 | Times |                    |
| Suspend delay during programming                                 | t <sub>SPD</sub>   | _    | _    | 120  | μS    | Figure             |
| 1st time suspend delay during erasing when suspend priority mode | t <sub>SESD1</sub> | _    | _    | 120  | μS    | <sup>-</sup> 32.38 |
| 2nd time suspend delay during erasing when suspend priority mode | t <sub>SESD2</sub> | _    | _    | 1.7  | ms    | _                  |
| Suspend delay during erasing when erasing priority mode          | t <sub>SEED</sub>  | _    | _    | 1.7  | ms    | _                  |

Note: \* User Boot Mat is 32 Kbytes.

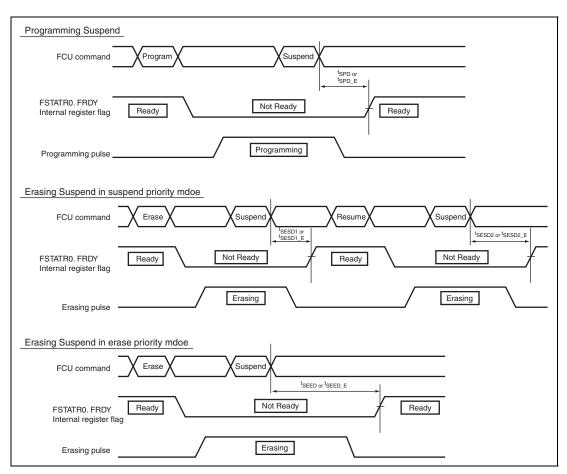


Figure 32.38 Flash Programming/Erasing Suspend Timing

#### 32.6 EEPROM Characteristics

Table 32.31 shows the EEPROM characteristics. The characteristics are measured under the condition that the frequency of the clock signal input on the EXTAL pin is 20 MHz.

#### **Table 32.31 EEPROM Characteristics**

Conditions: 
$$V_{cc} = PLLV_{cc} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}1 = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}, PV_{cc}2 = 5.0 \text{ V} \pm 0.5 \text{ V}, AVREFH_A = AVREFH_B = 4.5 \text{ V} \text{ to } AV_{cc}, V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 \text{ V} When  $PV_{cc}1 = 3.3 \text{ V} \pm 0.3 \text{ V}, V_{cc} = PV_{cc}1. T_a = -40^{\circ}\text{C} \text{ to } 125^{\circ}\text{C}$$$

| Item   |              |   | Symbol                           | Min. | Тур. | Max.   | Unit  | Figures            |
|--|--------------|---|----------------------------------|------|------|--------|-------|--------------------|
| Programming  | 8 bytes      |   | t <sub>P8_E</sub>                | _    | 0.4  | 2      | ms    |                    |
| time   | 128 bytes    |   | t <sub>P128_E</sub>              | _    | 1.0  | 5.0    | ms    | _                  |
| Erase time   | 8 Kbytes     | Programming/erasing: 100 times or less  | t <sub>E8K_E</sub>               | _    | 80   | 250    | ms    |                    |
|  |              | Programming/erasing: 101 to 50000 times |                                  | _    | 250  | 750    | _     |                    |
| Blank check  | 8 bytes      |   | t <sub>BC8_E</sub>               | _    | _    | 30     | μS    |                    |
| time   | 8 Kbytes     |   | t <sub>BC8K_E</sub>              |      | _    | 2.5    | ms    | <del>-</del>       |
| Reprogrammi  | ng/erasing c | ycle                                    | $N_{\scriptscriptstyle{PEC\_E}}$ |      | _    | 50000* | Times |                    |
| Suspend dela   | y during pro | gramming                                | t <sub>SPD_E</sub>               | _    | _    | 120    | μS    | Figure             |
| 1st time suspend delay during erasing when suspend priority mode |              |   | t <sub>SESD1_E</sub>             | _    | _    | 120    | μS    | <sup>-</sup> 32.38 |
| 2nd time suspend delay during erasing when suspend priority mode |              |   | t <sub>SESD2_E</sub>             | _    | _    | 1.7    | ms    | _                  |
| Suspend dela<br>when erasing                                     | , ,          | •                                       | t <sub>SEED_E</sub>              | _    | _    | 1.7    | ms    | _                  |

Note: \* In case that the programming/erasing cycle is 30000 times or less, the data retention period is 15 years.

In case that the programming/erasing cycle is 30001 to 50000 times, the data retention period is 2 years.

### 32.7 Usage Note

#### 32.7.1 Notes on Connecting External Capacitor for Current Stabilization

This LSI includes an internal step-down circuit to automatically reduce the microprocessor power supply voltage to an appropriate level. Between this internal stepped-down power supply ( $V_{\rm CL}$  pin) and the  $V_{\rm ss}$  pin, a capacitor for stabilizing the internal voltage. Connection of the external capacitor is shown in figure 32.39. The external capacitor should be located near the pin. Do not apply any power supply voltage to the  $V_{\rm CL}$  pin.

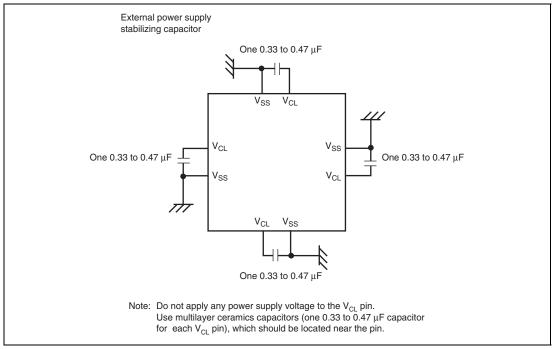


Figure 32.39 Connection of V<sub>CL</sub> Capacitor

## Table 32.32 Capacitance for Internal Voltage Stabilization

| Item   |         | Symbol                             | Min. | Max. | Unit |  |
|--|---------|------------------------------------|------|------|------|--|
| Capacitance for internal                     | Per pin | CV <sub>CL</sub>                   | 0.33 | 0.47 | μF   |  |
| voltage stabilization (V <sub>CL</sub> pins) | Total   | $\Sigma CV_{\scriptscriptstyleCL}$ | 1.32 | 1.88 | μF   |  |

# Appendix

### A. Pin States

Tables A.1 to A.3 show the pin states of SH72546R.

**Table A.1** Pin States

| Pin Function   |                     | Pin State          |                    |                               |                     |                     |  |  |  |
|----------------|---------------------|--------------------|--------------------|-------------------------------|---------------------|---------------------|--|--|--|
|                |                     |                    | Power-Down<br>Mode |                               |                     |                     |  |  |  |
|                |                     | ROM Disab          |                    |                               |                     |                     |  |  |  |
|                |                     | Mode               |                    |                               |                     |                     |  |  |  |
| Classification | Abbreviation        | 8 Bits             | 16 Bits            | ROM Enabled<br>Extension Mode | Single-Chip<br>Mode | Hardware<br>Standby |  |  |  |
| Clock          | CK                  | 0                  | 0                  | 0                             | 0                   | Z                   |  |  |  |
|                | XTAL                | 0                  | 0                  | 0                             | 0                   | L                   |  |  |  |
|                | EXTAL               | I                  | I                  | 1                             | I                   | I                   |  |  |  |
| System control | HSTBY               | Ţ                  | I                  | I                             | I                   | 1                   |  |  |  |
|                | RES                 | I                  | I                  | 1                             | I                   | Z                   |  |  |  |
|                | ASEMD               | I<br>(pulled down) | (pulled down)      | l<br>(pulled down)            | (pulled down)       | I                   |  |  |  |
|                | MD_CLKP             | I                  | I                  | 1                             | I                   | I                   |  |  |  |
|                | MD_CLK1,<br>MD_CLK0 | I                  | 1                  | 1                             | I                   | I                   |  |  |  |
|                | MD4                 | I                  | I                  | 1                             | I                   | I                   |  |  |  |
|                | MD3                 | ļ                  | I                  | 1                             | I                   | I                   |  |  |  |
|                | FWE                 | ļ                  | I                  | 1                             | I                   | I                   |  |  |  |
|                | MD2 to MD0          | I                  | I                  | 1                             | I                   | I                   |  |  |  |
|                | WDTOVF              | 0                  | 0                  | 0                             | 0                   | Z                   |  |  |  |
| Interrupt      | NMI                 | Ţ                  | I                  | I                             | I                   | Z                   |  |  |  |
|                | IRQ7 to IRQ0        | _                  | _                  | _                             | _                   | _                   |  |  |  |
| Address bus    | A21 to A0           | 0                  | 0                  | 0                             | _                   | Z                   |  |  |  |
| Data bus       | D7 to D0            | I (pulled up)      | I (pulled up)      | I (pulled up)                 | _                   | Z                   |  |  |  |
|                | D15 to D8           | _                  | I (pulled-up)      | I (pulled-up)                 | _                   | Z                   |  |  |  |
|                |                     |                    |                    |                               |                     |                     |  |  |  |

Pin Function Pin State Power-Down **Power-On Reset State** Mode **ROM Disabled Extension** Mode **ROM Enabled** Single-Hardware Classification Abbreviation 16 Bits 8 Bits **Extension Mode Chip Mode Standby** WAIT Ī Ζ Bus control ı ı WE<sub>1</sub> Н Н Z WE0 Н Z Н Н  $\overline{RD}$ Н Н Z Н RD/WR Н Н Н Z CS3 to CS1 CS0 Н Н Н Z POD Ports ATU-III TIA05 to TIA00 TIOC43 to TIOC40. TIOC33 to TIOC30. TIOC23 to TIOC20. TIOC13 to TIOC10. TIOC03 to TIOC00 TOD33A to TOD30A, TOD23A to TOD20A. TOD13A to TOD10A. TOD03A to

TOD00A

Pin Function Pin State Power-Down **Power-On Reset State** Mode **ROM Disabled Extension** Mode **ROM Enabled** Single-Hardware Classification Abbreviation 16 Bits 8 Bits Extension Mode Chip Mode Standby ATU-III TOD33B to TOD30B. TOD23B to TOD20B. TOD13B to TOD10B, TOD03B to TOD00B TOE53 to TOE50. TOE43 to TOE40. TOE33 to TOE30. TOE23 to TOE20. TOE13 to TOE10. TOE03 to TOE00 TIF2A to TIF0A TIF2B to TIF0B TIF19 to TIF3 TIJ1, TIJ0 TCLKA, TCLKB SCI SCK\_A to SCK\_E TxD\_A to  $TxD_E$ 

Pin Function Pin State

|                |                       |           | Power-On Reset State |                               |                      |                     |  |
|----------------|-----------------------|-----------|----------------------|-------------------------------|----------------------|---------------------|--|
|                |                       | ROM Disab | led Extension        |                               |                      |                     |  |
|                |                       | N         | lode                 | _                             |                      |                     |  |
| Classification | Abbreviation          | 8 Bits    | 16 Bits              | ROM Enabled<br>Extension Mode | Single-<br>Chip Mode | Hardware<br>Standby |  |
| SCI            | RxD_A to<br>RxD_E     | _         | _                    | _                             | _                    | _                   |  |
| RSPI           | RSPCKA to<br>RSPCKC   | _         | _                    | _                             | _                    | _                   |  |
|                | MOSIA to<br>MOSIC     | _         | _                    | _                             | _                    | _                   |  |
|                | MISOA to<br>MISOC     | _         | _                    | _                             | _                    | _                   |  |
|                | SSLA0 to<br>SSLC0     | _         | _                    | _                             | _                    | _                   |  |
|                | SSLA1 to<br>SSLA7     | _         | _                    | _                             | _                    | _                   |  |
|                | SSLB1 to<br>SSLB3     | _         | _                    | _                             | _                    | _                   |  |
|                | SSLC1 to<br>SSLC3     | _         | _                    | _                             | _                    | _                   |  |
| A/D converter  | AN_A27 to<br>AN_A0    | Z         | Z                    | Z                             | Z                    | Z                   |  |
|                | AN_B48 to<br>AN_B40   | Z         | Z                    | Z                             | Z                    | Z                   |  |
|                | ADTRG_A,<br>ADTRG_B   | _         | _                    | _                             | _                    | _                   |  |
|                | ADEND_A,<br>ADEND_B   | _         | _                    | _                             | _                    | _                   |  |
|                | AVREFH_A,<br>AVREFH_B | 1         | 1                    | I                             | I                    | I                   |  |
|                | AVREFL_A,<br>AVREFL_B | I         | 1                    | I                             | I                    | I                   |  |

Pin Function Pin State Power-Down **Power-On Reset State** Mode **ROM Disabled Extension** Mode **ROM Enabled** Single-Hardware Classification Abbreviation 16 Bits 8 Bits Extension Mode Chip Mode Standby **RCAN** CTx\_A to CTx\_C CRx\_A to CRx\_C **UBC UBCTRG** I/O ports PA15 to PA0 I Ζ PB14 ı Z ı 7 PB13 to PB11 ı ı PB10 to PB8 ı Ζ Z PB7 Ι ı PB6 ı Z Z PB5 to PB0 ı Z PC15 to PC8 ı ı PC7 to PC0 ı Z PD13 to PD0 ı I ı ı Z PE13 to PE0 ı ı ı ı Z ı PF15 to PF0 Ι Τ ı Z PG15 to PG0 ı I ı ı Ζ PH5 to PH0 ı ı Z Ι ı PJ9 to PJ0 ı ı ı Z Ι PK11 to PK0 I I ı I Ζ ı ı ı PL8 to PL0 ı 7

**Table A.2** Pin States for JTAG Interface

| Pin Fu         |              | Pin State    |          |                                      |                         |                     |                   |                          |
|----------------|--------------|--------------|----------|--------------------------------------|-------------------------|---------------------|-------------------|--------------------------|
|                |              |              | Power-O  | n Reset Sta                          | te                      | Power-Do            | wn Mode           |                          |
|                | •            | ROM Disabled |          |                                      |                         |                     |                   | -                        |
|                |              | Extens       | ion Mode |                                      |                         |                     |                   |                          |
| Classification | Abbreviation | 8 Bits       | 16 Bits  | ROM<br>Disabled<br>Extension<br>Mode | Single-<br>Chip<br>Mode | Hardware<br>Standby | Module<br>Standby | Left<br>Unconnect<br>-ed |
| JTAG           | TMS          | I            | I        | I                                    | I                       | Z                   | Z                 | Pulled up                |
|                | TRST         | 1            | I        | I                                    | I                       | Z                   | Z                 | Pulled up                |
|                | TDI          | I            | I        | 1                                    | I                       | Z                   | Z                 | Pulled up                |
|                | TDO          | O/Z          | O/Z      | O/Z                                  | O/Z                     | Z                   | Z                 | O/Z                      |
|                | TCK          | I            | I        | I                                    | I                       | Z                   | Z                 | Pulled up                |

Pin State

Table A.3 Pin States for Advanced User Debugger (AUD)

| FIII FU        | nction                   | FIII State          |                          |                             |                             |                             |                             |                          |  |  |
|----------------|--------------------------|---------------------|--------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|--------------------------|--|--|
|                |                          | Power-Down State    |                          | _                           | Reset                       | AUD in                      | Operation                   |                          |  |  |
| Classification | Abbreviation             | Hardware<br>Standby | AUD<br>Module<br>Standby | AUDMD<br>= H                | AUDMD<br>= L                | AUDMD<br>= H                | AUDMD<br>= L                | Left<br>Unconnect<br>-ed |  |  |
| AUD            | AUDRST                   | Z                   | Z                        | L input<br>(pulled<br>down) | L input<br>(pulled<br>down) | H input<br>(pulled<br>down) | H input<br>(pulled<br>down) | Pulled<br>down           |  |  |
|                | AUDMD                    | Z                   | Z                        | I (pulled<br>up)            | l (pulled<br>up)            | I (pulled<br>up)            | I (pulled<br>up)            | Pulled up                |  |  |
|                | AUDATA3<br>to<br>AUDATA0 | Z                   | Z                        | I (pulled<br>up)            | pulled up                   | I/O (pulled<br>up)          | 0                           | Pulled up                |  |  |
|                | AUDCK                    | Z                   | Z                        | I (pulled<br>up)            | pulled up                   | I (pulled<br>up)            | 0                           | Pulled up                |  |  |
|                | AUDSYNC                  | Z                   | Z                        | I (pulled<br>up)            | pulled up                   | I (pulled<br>up)            | 0                           | Pulled up                |  |  |

### [Legend]

—: Initial value undefined

I: Input O: Output

H: High level outputL: Low level outputZ: High impedance

Pin Function

Pulled up: Pulled up by the resistors inside the LSI

Pulled down: Pulled down by the resistors inside the LSI

## **B.** Product Code Lineup

## **Table B.1** Product Code Lineup

| Product Type | Product Code  | Package               |
|--------------|---------------|-----------------------|
| SH72546R     | R5F72546RKBGV | PRBG0272FA-A (BP-272) |

## C. Package Dimensions

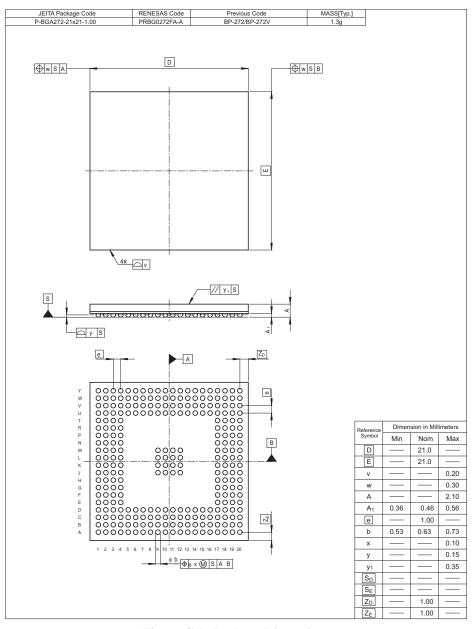


Figure C.1 Package Dimensions

## Main Revisions and Additions in this Edition

| Item              | Page | Revision (See Manual for Details)  |  |  |  |  |  |
|-------------------|------|--|--|--|--|--|--|
| _                 | _    | Terms amended  |  |  |  |  |  |
|                   |      | [Before amendment] $SCIA \rightarrow [After amendment] SCI\_A$   |  |  |  |  |  |
|                   |      | [Before amendment] $SCIB \rightarrow [After amendment] SCI\_B$   |  |  |  |  |  |
|                   |      | [Before amendment] $SCIC \rightarrow [After amendment] SCI\_C$   |  |  |  |  |  |
|                   |      | [Before amendment] $SCID \rightarrow [After amendment] SCI_D$  |  |  |  |  |  |
|                   |      | [Before amendment] $SCIE \rightarrow [After amendment] SCI_E$  |  |  |  |  |  |
|                   |      | [Before amendment] RSPIA $\rightarrow$ [After amendment] RSPI_A  |  |  |  |  |  |
|                   |      | [Before amendment] RSPIB $\rightarrow$ [After amendment] RSPI_B  |  |  |  |  |  |
|                   |      | Before amendment] $RSPIC \rightarrow [After amendment] RSPI\_C$  |  |  |  |  |  |
|                   |      | [Before amendment] RCANA $\rightarrow$ [After amendment] RCAN_A  |  |  |  |  |  |
|                   |      | [Before amendment] RCAN-A $\rightarrow$ [After amendment] RCAN_A   |  |  |  |  |  |
|                   |      | [Before amendment] RCANB $\rightarrow$ [After amendment] RCAN_B  |  |  |  |  |  |
|                   |      | [Before amendment] RCAN-B $\rightarrow$ [After amendment] RCAN_B   |  |  |  |  |  |
|                   |      | [Before amendment] RCANC $\rightarrow$ [After amendment] RCAN_C  |  |  |  |  |  |
|                   |      | [Before amendment] RCAN-C $\rightarrow$ [After amendment] RCAN_C   |  |  |  |  |  |
| _                 | _    | Pin name amended   |  |  |  |  |  |
|                   |      | [Before amendment] $RD\overline{WR} \rightarrow [After amendment] RD/\overline{WR}$                                      |  |  |  |  |  |
| 1.1 SH72546R      | 1    | Amended  |  |  |  |  |  |
| Features          |      | Note: F-ZTAT is a trademark of Renesas Electronics Corporation.  |  |  |  |  |  |
| Table 1.1         | 7    | Serial Communications interface (SCI), Features amended  |  |  |  |  |  |
| SH72546R Features |      | Maximum baud rate  |  |  |  |  |  |
|                   |      | Asynchronous: 1.25 Mbps  |  |  |  |  |  |
|                   |      | Synchronous: 2.5 Mbps  |  |  |  |  |  |
|                   |      | <ul> <li>One SCI channel for 1.25 Mbps operation is multiplexed with<br/>RCAN TL1 pins</li> </ul>                        |  |  |  |  |  |
|                   | 8    | Controller area Network (RCAN-TL1) , Features amended  |  |  |  |  |  |
|                   |      | Two or more RCAN-TL1 channels can be assigned to one bus to increase number of buffers with a granularity of 32 channels |  |  |  |  |  |
|                   |      |  |  |  |  |  |  |
|                   |      | <ul> <li>Parity error detection capability (1-bit parity based; every 8-bit<br/>message buffer)</li> </ul>               |  |  |  |  |  |

| Item                             | Page | Revision (See Manual for Details)  |  |  |  |
|----------------------------------|------|--|--|--|--|
| Figure 1.1 Block<br>Diagram      | 10   | Advanced (ATU-III) (WDT) (WDT) (MDT) |  |  |  |
|                                  |      | Peripheral controller (INTC) Port RES input ING input IN |  |  |  |
| Figure 1.2 Pin                   | 11   | Description added  |  |  |  |
| Arrangements                     |      | Top View   |  |  |  |
| 2.2 Register                     | 33   | Added  |  |  |  |
| Descriptions                     |      | The register set consists of five types of registers: sixteen 32-bit general registers, four 32-bit control registers, four 32-bit system registers, floating-point registers, and floating-point system registers.  |  |  |  |
| Section 3 Operating              | 83   | Added  |  |  |  |
| Section 3 Operating<br>Modes     |      | This LSI has a JTAG interface, so the TRST signal must be held low for a specified time when power is supplied. This is the case whether or not the JTAG interface is used, even in normal operating mode.   |  |  |  |
| 3.1 Types of                     | 83   | Added  |  |  |  |
| Operating Modes<br>and Selection |      | This LSI has six types of operating modes. The operating mode is determined by the setting of pins MD4 to MD0 and FWE. The setting should not be changed during LSI operation. However, the shift of mode from the MCU extended mode with the on-chip ROM enabled or the MCU single-chip mode to the user program mode is supported even when the FWE pin is operating. These pins should be set as shown in table 3.2.  |  |  |  |
| Table 3.2 Operating              | 84   | Pin name amended   |  |  |  |
| Mode Pin Settings                |      | [Before amendment] FEW $\rightarrow$ [After amendment] FWE   |  |  |  |
| Figure 4.1 Block                 | 86   | Direction of an arrow amended  |  |  |  |
| Diagram of CPG                   |      | [Before amendment] XTAL → Oscillator circuit   |  |  |  |
|                                  |      | ¥ [After amendment] XTAL ← Oscillator circuit  |  |  |  |

| Item   | Page | Revision (See Man  | ual for Details) |                |        |
|--|------|--|------------------|----------------|--------|
| Table 4.1 Pin  | 86   | Amended  |                  |                |        |
| Configuration  |      | Pin Name   | Symbol           | I/O            |        |
|  |      | External clock   | EXTAL            | Input          | }      |
|  |      | Crystal  | XTAL             | Output         |        |
|  |      | System clock   | CK               | Output         |        |
|  |      |  |                  |                |        |
| Table 4.5 Crystal                                    | 89   | Parameters amende  | ed               |                |        |
| Resonator<br>Parameters                              |      | Rs $\frac{Max.}{\Omega}$   |                  |                |        |
| (Recommended Values)                                 |      | Co <del>Max.</del> (pF)  |                  |                |        |
| 4.4.2 External Clock                                 | 89   | Amended  |                  |                |        |
| Input  |      | Leave the XTAL pin capacitance on the X                          | •                |                |        |
| Table 8.4 Interrupt                                  | 158  | Interrupt Source Nu  | mber amended     |                |        |
| Exception Handling<br>Vectors and Priority<br>Levels |      | [Before amendment  | ] ADMA → [After  | amendment]     | A-DMAC |
| 8.10.1 Timing to<br>Clear Interrupt<br>Source        | 174  | Added  |                  |                |        |
|  |      | To change interrupt least three times after and then execute the | er the dummy rea | d of the inter |        |

| Item  | Page | Revision (See Manual for Details)   |
|---|------|---|
| 9.5 Usage Notes   | 207  | Amended and deleted   |
|   |      | 1. The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the period from executing an instruction to rewrite the UBC register till the new value is actually rewritten, the desired break may not occur. In order to know the timing when the UBC register is changed, execute five or more NOP instructions after reading from the last written register. Instructions after then are valid for the newly written register value.         |
|   |      | :   |
|   |      | <ol> <li>Do not place a pre-execution break for the instruction<br/>immediately after a DIVU (or DIVS) instruction. Otherwise the<br/>break will be still taken even though an exception or interrupt<br/>occurs during the DIVU (or DIVS) instruction execution</li> </ol>   |
|   |      | 10. Do not place a post-execution break for a DIVU (or DIVS) instruction. Otherwise the break will be still taken even though an exception or interrupt occurs during the DIVU (or DIVS) instruction execution.   |
| Table 10.6 8-Bit<br>External Device<br>Access and Data<br>Alignment | 226  | Table title amended   |
| Table 11.1 Register   | 259  | DMAFR, Register Name amended  |
| Configuration   |      | DMA operation flag bit register   |
| 12.2.1 A-DMAC   | 321  | Note amended  |
| Operation Register (ADMAOR)   |      | If an NMI occurs, the A-DMAC enters the DMA-transfer-disabled state (for details, refer to section 12.3.3, Transfer Suspension and Resumption). To restart DMA transfer, use the NMIF bit in the DMA operation flag bit register (DMAFR) in the DMAC because the ADMAC does not support such a bit in the ADMAOR register (for details, refer to section 11, Direct Memory Access Controller (DMAC)). Clearing the NMIF bit in the DMAC enables DMA transfer in the A-DMAC. |

| Item   | Page | Revision (See Manual for Details)   |
|--|------|---|
| 12.3.2 Each A-   | 347, | Amended   |
| DMAC Channel<br>Operation  | 348  | <ol><li>Repeats operations in steps 2 and 3 until all the TV bits are<br/>cleared to 0.</li></ol>   |
| (4) Operation for A-<br>DMAC Channels<br>Used for RCAN                       |      | <ol> <li>Sets the TE bit to 1 if all the TV bits are cleared to 0 and if the<br/>specified transfer has been completed. In this case, if an interrupt<br/>is enabled (IE = 1), an interrupt is requested to the CPU.</li> </ol>   |
| <ul> <li>Operation         Details and         Transfer flow     </li> </ul> |      |   |
| Section 13   | —    | Amended   |
| Advanced Timer<br>Unit III (ATU-III)   |      | [Before amendment] PCLK $\rightarrow$ [After amendment] P $\phi$  |
| 13.10.4 Timer  | 408  | Bits 5 to 0, Description amended  |
| Status Register A (TSRA)   |      | [Setting condition]   |
| (TOTA)   |      | <ul> <li>When TCNTAn is transferred to ICRAn on assertion of the input<br/>capture signal (TIA0n)</li> </ul>  |
| 13.12 Overview of  | 421  | Amended   |
| Timer B  |      | The frequency-multiplied clock generator generates a clock signal by  |
| (2) Frequency-<br>Multiplied Clock<br>Generator                              |      | producing from 1 to 4095 cycles in response to an external-event input signal.  |
| Figure 13.12 Block<br>Diagram of Timer B                                     | 423  | Amended   |
|  |      | Event input  Clock bus Po   |
|  |      | Output from Block for correcting frequency-multiplied clock amended   |
|  |      | TCNT1D0 to TCNT1D3, TCNT2D0 to TCNT2D3  |
| 13.13.22 Correcting<br>Counter Clearing<br>Register B<br>(TCCLRB)            | 446  | Amended TCCLRB is constantly compared with TCNTB5. When they match, TCNTB5 is stopped and a counter clearing trigger is output to timer D. TCNT1Dn and TCNT2Dn in timer D are separately cleared by setting the corresponding counter clearing enable bit in timer control register Dn (TCRDn). |

| Item   | Page                | Revision (See Manual for Details)   |
|--|---------------------|---|
| Figure 13.13 Input-<br>Capture and<br>Compare-Match<br>Operation of<br>TCNTB0  | 447                 | Values of a register amended TCNTB0: [Before amendment] 05676 → [After amendment] 45676 [Before amendment] 0554 → [After amendment] 5554  |
| 13.14.3 Frequency-<br>Multiplied Clock<br>Signal Corrector   | 454                 | :  TCNTB3 is a 20-bit up-counter that is driven by the external event input (AGCK). On the selected transition of the AGCK signal, the value in TCNTB3 is transferred to TCNTB4, after which TCNTB3 is incremented by the value in the PIM bits.  :  As long as its value is lower that that in TCNTB4, TCNTB5 is incremented until it reaches the value in correcting counter clearing register B (TCCLRB). Incrementation of TCNTB5 then stops, regardless of the relation between its value and that of TCNTB4. In addition, counters of timer D (TCNT1Dn and TCNT2Dn) can be separately cleared by this match as a trigger when the corresponding counter clearing enable bit (C1CEDn/C2CEDn) in timer control register Dn (TCRDn) is set to 1. |
| Figure 13.20 TCNTB5 Operation (at Start-Up) Figure 13.21 Operation of TCNTB5 (with Correction at End of Cycle) Figure 13.22 Operation of TCNTB5 (without Correction at End of Cycle) | 456,<br>457,<br>458 | Timer counter name amended  [Before amendment] TCNTDn → [After amendment] TCNT1Dn  Added  Note: When PIM = H'020 (multiplying by 32) and TCCLRB = H'00080  This figure shows the case of TCNT1Dn. This is the same for TCNT2Dn.   |

| Item   | Page | Revision (See Manual for Details)  |
|--|------|--|
| 13.16.3 Timer  | 465  | Bits 7 to 4, Description amended   |
| Control Registers<br>C0 to C4 (TCRC0 to<br>TCRC4)          |      | and forced compare match occurs. At this time, a counter is cleared and a signal is output on pin TIOCnm according to the value in the IOCn <sup>0</sup> bits. TCNTCn restarts   |
|  | 466  | Bit 3, Description amended   |
|  |      | In PWM mode, do not set GRCn0 to GRCn3 to H'000000. If GRCn0 is set to H'000000, compare match occurs at illegal cycles.   |
| 13.18 Overview of  | 484  | Amended  |
| Timer D  |      | <ul> <li>Interrupt requests can be output on counter overflow. Four<br/>outputs from TCNT1Dn and four outputs from TCNT2Dn.</li> </ul>   |
| 13.18.1 Block  | 484  | Amended  |
| Diagram of Timer D   |      | Each subblock consists of two timer counters (TCNT1Dn and TCNT2Dn), one offset base register (OSBRDn), four output compare registers (OCRDnm), four general registers (GRDnm), four timer down counters (DCNTDnm), and controller. Each channel includes |
|  |      |  |
| 13.19.1 Timer Start  | 486  | Amended  |
| Register (TSTRD)   |      | TSTRD is an 8-bit readable/writable register that enables and disables two timer counters (TCNT1Dn and TCNT2Dn) and timer down counters (DCNTDnm) in subblocks D0 to D3. When the counter Dn start bit and   |
|  |      | Bits 3 to 0, Description amended   |
|  |      | at which this bit is set. Therefore, the time from when this bit is set to when TCNT1Dn and TCNT2Dn are incremented for the first time is less than the cycle of the clock of TCNT1Dn and TCNT2Dn.   |
| 13.19.2 Timer  | 487  | Amended  |
| Control Registers<br>D0 to D3 (TCRD0 to<br>TCRD3)          |      | from clock-bus lines 0 to 5. These registers also enable and disable capture of the timer offset base register and counter clearing requests output from timer B for TCNT1Dn and TCNT2Dn.  |
| 13.19.3 Timer I/O  | 492  | Bits 7 to 0, Description amended   |
| Control Registers<br>1D0 to 1D3<br>(TIOR1D0 to<br>TIOR1D3) |      | When these bits are set to B'00, compare match between OCRDnm and timer counter 1 (TCNT1Dn <del>m</del> ) is not performed. Otherwise, the compare match is performed. When the CMEADnm bit in   |

| Item  | Page | Revision (See Manual for Details)   |
|---|------|---|
| 13.19.4 Timer I/O<br>Control Registers<br>2D0 to 2D3<br>(TIOR2D0 to<br>TIOR2D3) | 493  | Deleted   |
|   |      | When TIOR2Dn is used as the compare match register (IOBDnm[2] = 0) and the IOBDnm[1:0] bits are set to B'00, compare match between GRDnm and timer counter 2 (TCNT2Dn <del>m</del> ) is not performed. Otherwise, the compare match is performed. When the CMEBDnm bit in   |
| 13.19.11 Timer  | 510  | Amended   |
| Output Control Registers D0 to D3 (TOCRD0 to TOCRD3)                            |      | Signals on pins TODnmA and TODnmB are inverted on the first edge of the $P\phi$ clock after the output inversion select Dn bit is set to 1. This function is not affected by the operating state of timer counters 1Dn and 2Dn (TCNT1Dn, TCNT2Dn).  |
| 13.20 Operations of   | 519  | Amended   |
| Timer D   |      | To set the clocks for TCNT1Dn, TCNT2Dn, or DCNTDnm or values in registers such as DCNTDnm, OCRDnm, and GRDnm while TCNT1Dn or TCNT2Dn is in operation, note that the value to be set may lead to malfunction. For example, while setting the compare match value, the counter value may exceed the value to be set.   |
| Figure 13.35 Block  | 546  | Legend amended  |
| Diagram of<br>Subblocks of Timer<br>F   |      | [Before amendment] Event counter $BF \rightarrow$ [After amendment] Event counter $F$   |
| 13.25.3 Timer   | 554  | Bits 1, 0, Description amended  |
| Control Registers F0 to F19 (TCRF0 to   |      | 00: Edge detection disabled   |
| TCRF19)   |      | 01: Rising edge   |
|   |      | 10: Falling edge  |
|   |      | 11: Both edges  |
| 13.25.5 Timer   | 556  | Term amended  |
| Status Registers F0<br>to F19 (TSRF0 to<br>TSRF19)                              |      | [Before amendment] TSRF00 $\rightarrow$ [After amendment] TSRF0   |
| 13.25.12 General<br>Registers DF12 to<br>DF15 (GRDF12 to<br>GRDF15)             | 566  | Amended   |
|   |      | GRDF12 to GRDF15 are 32-bit readable/writable registers. This register is provided to subblocks F12 to F15. Triggered by edge input of the TIFn pin, accumulated number in the time counter CFn (ECNTCFn) is taken in at the next ECNTAFn upcount timing. This register is valid only in 'rotation speed/pulse measurement' mode. Capture operation is not executed in other modes. |

| Item   | Page       | Revision (See Manual for Details)  |
|--|------------|--|
| 13.26.3<br>Measurement of<br>Time during<br>High/Low Input<br>Levels   | 577        | ECNTAFn: Executes upcount using one of the clock buses 0 to 5 as a count source and TIFnA level as enable. Therefore, the time period in which TIFnA is in high level is measured. After detecting a compare match in ECNTBFn, this register clears the count number synchronizing with the next count source clock. If TIFnA is driven high at clearing count by the compare match, the count value becomes H'00000100. Figure 13.40 is an example of this. |
| 13.27.2 Interrupt<br>Requests  | 589<br>611 | Amended and added Six timer G interrupts, CMIG0 to CMIG5, are available in timer G. When a compare match is detected in the subblocks, an interrupt request is output. The interrupt request is received in the direct memory access controller (DMAC) and interrupt controller (INTC).  Note 2 added  |
| Control Registers J0<br>and J1 (FCRJ0 and<br>FCRJ1)  | 011        | Note 2 added   |
| Figure 13.51<br>Operation Example<br>of Timer J (2)  | 624        | Value of data amended Captured data: [Before amendment] $5\ 4\ 5 \rightarrow$ [After amendment] $40\ 4\ 5$   |
| 13.36.3<br>Load/Reload<br>Conflict Operation<br>(1) Conflict between<br>Data Transfer and<br>Writing to Transfer<br>Destination Register | 641        | Amended  A conflict between data transfer between registers and a peripheral bus write to the transfer destination register is described below.  |
| (2) Conflict between<br>Data Transfer and<br>Writing to Transfer<br>Source Register  | 642        | Amended  A conflict between data transfer between registers and a peripheral bus write to the transfer source register is described below. When data transfer occurs   |
| Table 13.40 Resources Related to Conflict Operation between Counter Clearing and Compare Match   | 657        | Counter Clearing Source amended TCNT1Dn/TCNT2Dn clearing request from timer B  |

| Item  | Page          | Revision (See Manual for Details)   |
|---|---------------|---|
| (2) Conflict between<br>Writing to<br>TCNT1Dn/TCNT2D<br>n Counter and<br>Counter Clearing by<br>Timer B | 658           | Title amended   |
| 14.3.4 Watchdog   | 672           | Bit 7, Description amended  |
| Reset Control Register (WRCR)   |               | Note: This LSI is not initialized internally, but WTCNT and WTCR are initialized by an overflow of WTCNT. |
| Section 16 Serial   | _             | Term amended  |
| Communications<br>Interface (SCI)   |               | [Before amendment] SCIF $\rightarrow$ [After amendment] SCI   |
| Figure 16.5 Example<br>of Transmission in<br>Asynchronous Mode<br>(8-Bit Data, Parity,<br>One Stop Bit) | 723           | Figure replaced   |
| Figure 16.11<br>Example of SCI<br>Transmit Operation  | 731           | Figure replaced   |
| Section 17 Renesas<br>Serial Peripheral<br>Interface (RSPI)   | _             | Descriptions including "SH72546R" are modified to descriptions using "LSI".                               |
| Table 17.2 Register Configuration   | 748 to<br>750 | Access Sizes of registers SSLPA/B/C, SPSRA/B/C, SPSSRA/B/C and SSLNDA/B/C amended                         |
|   |               | [Before amendment] 8, 16 $\rightarrow$ [After amendment] 8  |
| Figure 17.14 SPTEF<br>and SPRF Bit<br>Operation Example   | 790           | Figure replaced   |
| Figure 18.2 RCAN-<br>TL1 Memory Map   | 820           | Reserved area between Mailbox-31 and MBESR added  |
| 18.3.1 Memory Map   | 820           | Amended   |
|   |               | The locations not used (between H'000 and H'602) are reserved and cannot be accessed.                     |
| Figure 18.3 Mailbox-<br>n Structure   | 824 to<br>826 | Figure title amended  |
| Figure 18.3 Mailbox-<br>n Structure<br>(continued)  |               |   |

| Item  | Page | Revision (See Manual for Details)   |
|---|------|---|
| Figure 18.4                                 | 831  | Mailbox number amended  |
| Acceptance filter                           |      | [Before amendment] $N \rightarrow$ [After amendment] $n$  |
| (4) Timestamp                               | 834  | Deleted   |
|   |      | Important: Please note that the TimeStamp is stored in a temporary register. Only after a successful transmission or reception the value is then copied into the related Mailbox field. The TimeStamp may also be updated if the CPU clears RXPR[N]/RFPR[N] at the same time that UMSR[N] is set in overrun, however it can be read properly before clearing RXPR[N]/RFPR[N]. |
| 18.4.3 Message                              | 902  | Amended and added   |
| Transmission<br>Sequence                    |      | TTT (Mailbox-n) – TTT (Mailbox-n-1) > TEW + the maximum frame length + 9  |
| (1) Time Triggered<br>Transmission          |      | n = 25 to 30  |
| Setting Tx-     Trigger Time                |      |   |
| Figure 18.24<br>Message receive<br>sequence | 912  | Figure replaced   |
| 18.4.4 Message                              | 913  | Amended   |
| Receive Sequence                            |      | takes place at Mailbox-30 (if configured as receive). Once RCAN-TL1 finds a matching identifier, it stores the number of Mailbox-[n] into an internal buffer, stops the search, and goes back to idle state, waiting for the EndOfFrame (EOF) to come. When the 6 <sup>th</sup> bit of EOF is   |
| Table 18.2 RCAN-                            | 917  | Amended   |
| TL1-n* <sup>1</sup> Interrupt<br>Sources    |      | Notes: 2. RM0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1 is an interrupt generated by the remote request pending flag for mailboxes 1 to 31 (RFPR1/RFPR0) or the data frame receive flag for mailboxes 1 to 31 (RXPR1/RXPR0).   |
| 18.7 DMAC                                   | 918  | Amended   |
| Interface                                   |      | The DMAC can be activated by the reception of a message in RCAN-TL1 mailbox 0. When DMAC transfer ends after DMAC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-TL1 cannot be sent to the CPU in this case. Figure 18.27 shows a DMAC transfer flowchart.                        |

| 919           | Amended   |  |
|---------------|---|--|
|               | 7111011404  |  |
|               | A Renesas HA13721 transc  | essary to connect this LSI to a CAN bus. eiver IC and its compatible products are shows a sample connection diagram.   |
| 932 to<br>934 | Note 2 added  |  |
| 949,<br>950   | Note added  |  |
| 962 to<br>964 | Note added  |  |
| 968           | Module name amended [Before amendment] SCIF -                                       | → [After amendment] SCI  |
| 980           | Deleted   |  |
|               | is selected (CKS = 0), the tir<br>1 is 4 $\overline{\text{(TBD)}}$ states after the | nal of trigger input timing. If the low-speed ming at which the ADSCACT bit is set to falling edge of the ADTRG pin is seed is selected (CKS = 1), it is 3 (TBD) he ADTRG pin is sampled.  |
| 987 to<br>990 | Replaced  |  |
| _             | Deleted   |  |
| 994           | Deleted   |  |
|               | Abbreviation  | Initial Value* <sup>1</sup>  |
|               | SDBPR   | Undefined* <sup>2</sup>  |
|               | SDBSR   | Undefined <del>≚<sup>e</sup></del>   |
| 9 9 9 9       | 62 to 68 80 87 to   | Note 2 added  49, Note added  62 to Note added  64  Module name amended [Before amendment] SCIF –  80 Deleted Figure 19.9 shows an exterr is selected (CKS = 0), the tir 1 is 4 (TBD) states after the sampled; while if the high-sp states after the fall edge of t  87 to Replaced  90  Deleted  Abbreviation  SDBPR |

| Item                                | Page          | Revision (See Manual for Details)   |
|-------------------------------------|---------------|---|
| 20.3.3 Bypass                       | 995           | Deleted   |
| Register (SDBPR)                    |               | SDBPR is a 1-bit register and is connected between the TDI and TDO pins when SDIR is set to BYPASS mode. The initial value is undefined. The value in SDBPR is not initialized by a power-on reset or TRST pin assertion, but the first value read by the BYPASS instruction is always 0 because this register is initialized in the Capture-DR state of this instruction.  |
| 20.5 Usage Notes                    | 1016          | Amended and added   |
|                                     |               | 3. Whether or not the JTAG interface is in use, be sure to hold the TRST signal at the low level when resetting the chip. Specifically, hold the TRST signal low for 20 TCK clock cycles. For details, see section 32, Electrical Characteristics.  |
|                                     |               | <ol> <li>Up to 2 MHz frequency can be input to the TCK. Normal operation<br/>cannot be guaranteed if the frequency exceeds 2 MHz. For<br/>details, see section 32, Electrical Characteristics.</li> </ol>   |
| Table 23.1 Register                 | 1146,<br>1147 | Initial Values amended  |
| Configuration                       |               | [Before amendment] $H'xxxx \rightarrow [After amendment]$ Pin state   |
| 23.4.4 Port B Driving               | 1157          | Amended   |
| Ability Setting<br>Register (PBDSR) |               | PBDSR is a 16-bit readable/writable register that selects the driving ability for port B. Bits PB13DSR, PB12DSR, and PB4DSR to PB0DSR correspond to pins PB13/CS3/RSPCKB, PB12/CS2/RSPCKA, and PB4/A20/CTx_B/TIF6 to PB0/A16/MOSIA, respectively. The PBDSR setting is ignored when the corresponding pin function is specified as bus function (A16 to A20, CS2, and CS3). |

| Item  | Page                    | Revision (See Manual for Details)  |
|---|-------------------------|--|
| 23.4.4 Port B Driving   | 1157,<br>1171,<br>1177, | Descriptions amended   |
| Ability Setting<br>Register (PBDSR)   |                         | [Before amendment]   |
| 23.7.4 Port E Driving<br>Ability Setting  | 110/                    | When these bits are set to 1, the driving ability of the corresponding pin is higher than normal.  0: Normal driving ability   |
| Register (PEDSR)  |                         | High driving ability   |
| 23.8.4 Port F Driving<br>Ability Setting<br>Register (PFDSR)                                |                         | [After amendment]  |
| 23.9.4 Port G<br>Driving Ability<br>Setting Register  |                         | When these bits are set to 1, the driving ability of the corresponding pin is higher than normal.  |
| (PGDSR)   |                         | 0: Normal driving ability (slow slew rate)   |
| 23.11.4 Port J  |                         | 1: High driving ability (fast slew rate)   |
| Driving Ability<br>Setting Register<br>(PJDSR)  |                         | Note: For the characteristics of the driving ability, refer to section 32.3.14, Output Slew Rate.  |
| 23.12.4 Port K<br>Driving Ability<br>Setting Register<br>(PKDSR)                            |                         |  |
| 23.9.5 Port G Edge  | 1186                    | Deleted  |
| Selecting Register (PGER)   |                         | Note: * Since edge detection is not performed, bits in the port G data register retain 0.  |
| 23.11.3 Port J  | 1193                    | Amended  |
| Inverting Register (PJIR)   |                         | Setting the specific bits of PJIR to 1 inverts the output value on their corresponding pins.   |
| 23.16.2 Note on<br>Operation of<br>Input/Output Pins on<br>a Reset by an<br>Internal Source | 1211                    | Added  |
| Section 25 ROM  | _                       | Descriptions including "SH72546R" are modified to descriptions using "LSI".  |
| 25.3.5 ROM MAT  | 1230                    | Added  |
| Select Register<br>(ROMMAT)   |                         | When switching MAT by changing ROMMAT register setting, write to ROMMAT register and execute a minimum of 5 NOP instructions after performing a dummy read of ROMMAT register. |

| Item   | Page          | Revision (See Manual for Details)   |  |
|--|---------------|---|--|
| 25.3.9 FCU RAM<br>ECC Error Control<br>Register<br>(FRAMECCR)    | 1238          | Amended and deleted   |  |
|  |               | Indicates that a 1-bit error has been corrected when the FCU is reading RAM.  |  |
|  |               | 0: Issues no FCU command-lock request upon detection of a 1-bit error.  |  |
|  |               | 1: Issues an FCU command-lock request upon detection of a 1-bit error.  |  |
| 25.3.10 Flash P/E  | 1239          | Added   |  |
| Mode Entry Register (FENTRYR)                                    |               | When transiting to ROM read mode by changing FENTRY5 to FENTRY0 bits in FENTRYR register from 1 to 0, write 0 to FENTRY5 to FENTRY0 bits and execute a minimum of 5 NOP instructions after performing a dummy read of FENTRYR register. |  |
| 25.4 Overview of   | 1252          | Deleted   |  |
| ROM-Related<br>Modes   |               | The user boot MAT can be programmed or erased only in boot mode.  |  |
|  |               | In boot mode, the user MAT, user boot MAT, and EEPROM data  |  |
|  |               | MAT are all crased immediately after the LSI is started. The user-  |  |
|  |               | MAT, user boot MAT, and data MAT can then be programmed   |  |
|  |               | from the host via the SCI. The ROM can also be read after this-   |  |
|  |               | entire area erasure.  |  |
|  |               | In user boot mode, a boot operation with a desired interface can  |  |
|  |               | be implemented through mode pin settings different from those in  |  |
|  |               | user program mode.  |  |
| 25.5.2 State<br>Transition in Boot<br>Mode                       | 1255,<br>1256 | Note added  |  |
| (4) Waiting for Host<br>Command for<br>Programming or<br>Erasure |               |   |  |
| 25.5.4<br>Inquiry/Selection<br>Host Command Wait<br>State        | 1260          | Amended   |  |
|  |               | Product code (n bytes): ASCII code for the name of the supported device   |  |
| (1) Supported<br>Device Inquiry                                  |               |   |  |

| Item   | Page | Revision (See Manual for Details)   |  |  |
|--|------|---|--|--|
| (9) Erasure Block<br>Information Inquiry                     | 1268 | Appearance of Response amended  |  |  |
|  |      | Amended   |  |  |
|  |      | Size (2 bytes): Total number of bytes in the block count, block start address, and block end address fields   |  |  |
| (11) New Bit Rate  | 1272 | Amended   |  |  |
| <ul><li>Selection</li><li>Bit rate selection error</li></ul> |      | A bit rate selection error occurs when the bit rate selected through a new bit rate selection command cannot be set for the SCI of this LSI within an error of 4%. The bit rate error can be obtained by the following equation from the bit rate (B) selected through a new bit rate selection command, the input frequency (fEX), multiplication ratio 2 (MP $\varphi$ ), the SCBR1A setting (N) in SCI_A, and the CKS[1:0] bit value (n) in SCSMR1A. |  |  |
|  |      | Error (%) = $\left\{ \begin{array}{c} f_{EX} \times M_{P\phi} \times 10^6 \\ \hline (N+1) \times B \times 32 \times 2^{2n-1} \end{array} - 1 \right\} \times 100$   |  |  |
| (13) Key Code  | 1273 | Amended   |  |  |
| Check Command  |      | When receiving a key code check command, this LSI compares the key code sent from the host and the one in the MAT. When the key codes match, this LSI returns a response (H'26), erases the entire area of each of the user MAT, user boot MAT, and data MAT, then enters the programming/erasure state. When the key codes do not match, this LSI returns an error response (H'E0).  |  |  |
|  |      | When the key code of this LSI is in the initial state (meaning nothing has been written), this LSI returns a response (H'26) to any key code, erases the entire area of each of the user MAT, user boot MAT, and data MAT, then enters the programming/erasure state.   |  |  |
|  |      | [Legend]  |  |  |
|  |      | Key code (16 bytes): Key code for boot mode approval; pad higher-order bits that are not required with H'FF.  |  |  |
|  |      | H'51: Erasure cannot be done due to an erasure error  |  |  |
| (14) Boot Program<br>Status Inquiry                          | 1274 | Added   |  |  |
|  |      | Command H'4F  |  |  |
|  |      | Response H'5F Size Status Error SUM   |  |  |
|  |      | [Legend]  |  |  |
|  |      | SUM (1 byte): Checksum  |  |  |

| Item  | Page          | Revision (See Manual for Details)   |   |
|---|---------------|---|---|
| Table 25.8 Error<br>Code  | 1275          | H'54 deleted  |   |
| 25.5.5<br>Programming/Erasin<br>g Host Command<br>Wait State<br>(8) User MAT<br>Checksum        | 1283          | Added MAT checksum (4 bytes):   | Checksum of the user MAT data The user MAT also stores the key code for boot mode/debugging function authentication. Note that the checksum includes this key code value.  The indicated checksum is for 4 Mbytes. Since the user MAT of this LSI is 3.75 Mbytes, the unused 0.25 Mbytes are added as H'FF in the indicated checksum. |
| 25.5.5 Programming/Erasin g Host Command Wait State (9) User Boot MAT Blank Check (10) User MAT | 1284          | the area where the data is u<br>(e.g., reset input, power-sup   | rovided to check program/erase state of undefined by suspend of program/erase oply interruption). Therefore, if the sed again, make sure to completely erase  |
| Blank Check (11) Read Lock Bit Status (12) Lock Bit Program                                     | 1285,<br>1286 | Format of Legend amended  | <u>.                                    </u>  |
| 25.6.2 Conditions for FCU Command Acceptance (1) ROM Read Mode  ROM/EEPROM read mode            | 1292          | Amended and added  The ROM and EEPROM can be read through the ROM cache and peripheral bus, respectively, at a high speed. The FCU does not accept commands. The FCU enters this mode when the FENTRY5 to FENTRY0 bits in FENTRYR are set to 000000 and the FENTRYD bit to 0 in FENTRYR.  When transiting to ROM read mode by changing FENTRY5 to FENTRY0 bits in FENTRYR register from 1 to 0, write 0 to FENTRY5 to FENTRY0 bits and execute a minimum of 5 NOP instructions after performing a dummy read of FENTRYR register.  Table replaced |   |
| Modes/States and<br>Acceptable<br>Commands  |               | . 3310 10010000   |   |

| Item  | Page | Revision (See Manual for Details)  |  |
|---|------|--|--|
| 25.6.3 FCU  | 1296 | Amended and added  |  |
| Command Usage   |      | The FRDY bit retains 0, if the FRDTCT and FRCRCT bits are set to 1 to put the FCU into a command-locked state in the middle of its command handling while the FCUERR bit is 1 or the FRDCLE and FRCCLE bits are 1. Since the FCU in a command-locked state halts its processes, the FRDY bit is never set to 1 from 0. If the FRDY retains 0 for a longer period than programming/erasing time or suspend delay time (see section 32, Electrical Characteristics), abnormal operation such as the FCU process halt may have occurred. In such case, initialize the FCU by an FCU reset. If the FRDY is set to 1 upon completion of the FCU command handling, while the FRDCLE and FRCCLE bits are 1, the FCUERR bit and the FRDTCT and FRCRCT bits are also 0. Therefore, the state of error occurrence can be checked via the ILGLERR, ERSERR, and PRGERR bits. |  |
| 25.9.1 Switching<br>between User MAT<br>and User Boot MAT | 1324 | Amended  |  |
|   |      | <ol> <li>Flushing the ROM cache during MAT switching         Flush all lines of the ROM cache by writing 1 to the RCF bit in         RCCR (see section 27, ROM Cache (ROMC)).</li> <li>Added</li> </ol>  |  |
|   |      | <ol> <li>Executing five or more NOP instructions</li> <li>After the dummy read of ROMMAT, execute five or more NOP instructions.</li> </ol>  |  |
| Figure 25.32<br>Example of MAT<br>Switching Steps         | 1324 | Figure replaced  |  |
| 25.9.2 Other Notes  | 1327 | Added  |  |
| (9) Power off during<br>Programming or<br>Erasure         |      |  |  |
| (10) Prohibition of<br>Clearing FRDCLE<br>Bit to 0        |      |  |  |
| (11) Note on<br>Transition to ROM<br>Read Mode            |      |  |  |
| Section 26<br>EEPROM                                      | _    | Descriptions including "SH72546R" are modified to descriptions using "LSI".  |  |

| Item  | Page | Revision (See Manual for Details)  |
|---|------|--|
| 26.1 Features   | 1332 | Added  |
| Blank check<br>function   |      | Blank check function checks the erase state of the area where erase has ended. The function is disabled when programming/erasing was suspended (e.g., reset input, power-supply interruption).   |
| 26.3.9 EEPROM<br>Blank Check Control<br>Register<br>(EEPBCCNT)                    | 1348 | Title amended  |
| 26.4 Overview of<br>EEPROM-Related<br>Modes                                       | 1352 | ◆ In boot mode, the user MAT and user boot MAT in the ROM and the data MAT are all crased immediately after the LSI is started.  The data MAT can then be programmed from the host via the SGI. The data MAT can also be read after this entire area.                          |
|   |      | <ul> <li>erasure.</li> <li>In user boot mode, a boot operation with a desired interface can<br/>be implemented through mode pin settings different from those in<br/>user mode or user program mode.</li> </ul>  |
| 26.5.2<br>Programming/Erasin<br>g Host Commands<br>(1) Data MAT<br>Checksum       | 1357 | Legend amended Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4) MAT checksum (4 bytes): Checksum of the data MAT data SUM (1 byte): Checksum (for the response data)  |
| 26.6.2 Conditions for<br>FCU Command<br>Acceptance<br>(2) ROM/EEPROM<br>Read Mode | 1361 | Amended  The EEPROM can be read through the peripheral bus, and the ROM can be read through the ROM cache at a high speed. The FCU does not accept commands. The FCU enters this mode when the FENTRY5 to FENTRY0 bits are set to 0000000 and the FENTRYD bit to 0 in FENTRYR. |
| Table 26.8 FCU<br>Modes/States and<br>Acceptable<br>Commands                      | 1363 | Table replaced   |
| Figure 26.6 Procedure for EEPROM Programming                                      | 1365 | Term amended [Before amendment] $t_{_{P128}} \rightarrow$ [After amendment] $t_{_{P128}}$  |
| Figure 26.7<br>Procedure of the<br>EEPROM Blank<br>Check                          | 1367 | Term amended [Before amendment] $t_{\text{\tiny BCBK}} \rightarrow$ [After amendment] $t_{\text{\tiny BCBK}\_E}$   |

| Item   | Page | Revision (See Manual for Details)   |  |  |  |  |  |
|--|------|---|--|--|--|--|--|
| 26.9 Usage Notes                                   | 1373 | Added   |  |  |  |  |  |
| (5) Reset during<br>Programming or<br>Erasure      |      | No verification function is provided to check program/erase state of the area where the data is undefined by suspend of program/erase (e.g., reset input, power-supply interruption). Therefore, if the undefined area should be used again, make sure to completely erase data before usage.     |  |  |  |  |  |
| (9) Power off during<br>Programming or<br>Erasure  | 1374 | Added   |  |  |  |  |  |
| (10) Prohibition of<br>Clearing FRDCLE<br>Bit to 0 |      |   |  |  |  |  |  |
| 27.3.1 ROM Cache                                   | 1380 | Bit 0, Description amended  |  |  |  |  |  |
| Control Register (RCCR)                            |      | 0: The ROM cache function is not used.  |  |  |  |  |  |
| (HOOH)   |      | 1: The ROM cache function is used.  |  |  |  |  |  |
| 27.3.2 ROM Cache                                   | 1383 | Bit 0, Description amended  |  |  |  |  |  |
| Control Register 2 (RCCR2)                         |      | 0: The data cache function is not used.   |  |  |  |  |  |
| (1100112)  |      | 1: The data cache function is used.   |  |  |  |  |  |
| Section 28 RAM                                     | 1389 | Added   |  |  |  |  |  |
|  |      | This LSI incorporates 128-Kbyte RAM, which is connected to F (CPU instruction Fetch), M (Memory access), and I (Internal) buses. This on-chip RAM can be accessed via any of these buses independently.   |  |  |  |  |  |
| 28.1 Features                                      | 1389 | Amended   |  |  |  |  |  |
|  |      | • Access  |  |  |  |  |  |
|  |      | :   |  |  |  |  |  |
|  |      | For reading from the on-chip RAM, 1 or 2 cycles should be specified by a register according to the operating frequency (PLL multiplication ratio). For writing to the on-chip RAM, 2 or 3 cycles must be specified by a register according to the operating frequency (PLL multiplication ratio). |  |  |  |  |  |
|  |      | • Ports   |  |  |  |  |  |
|  |      | Each page in the on-chip RAM has two independent read and write ports. The read port is connected to I, F, and M buses and the write port is connected to I and M buses. The F and M buses are used for accesses from the CPU. The I bus is used for accesses from other than the CPU.            |  |  |  |  |  |

| Item | Page | Revision (See Manual for Details) |
|------|------|-----------------------------------|
|      |      |                                   |

28.2.1 RAM Enable Control Registers 0 and 1 (RAMEN0 and RAMEN1)

1393 Amended and deleted

If the RAME15 to RAME0 bits corresponding to the access page are set to 1, accessing the on-chip RAM becomes enabled; while if the RAME bits are cleared to 0, the on-chip RAM cannot be accessed. In the access disabled state, an undefined data is read if the page is read or if an instruction in the page is fetched, and a write to the page is ignored. The initial values of the RAME bits are 1.

To rewrite to the RAME15 to RAME0 bits in RAMEN0, word size data with upper byte as H'96 and lower byte as write data must be written. Similarly, to rewrite to RAME15 to RAME0 bits in RAMEN1, word size data with upper byte as H'95 and lower byte as write data must be written.

:

1404

To rewrite the RAME15 to RAME0 bits in RAMEN0, execute an instruction to read from RAMEN0 together with five or more NOP instructions immediately after the instruction to write to RAMEN0. To rewrite the RAME15 to RAME0 bits in RAMEN1, execute instructions for RAMEN1 in the same way as for RAMEN0.

28.2.2 RAM Write Enable Control Registers 0 and 1 (RAMWEN0 and RAMWEN1) Amended

If the RAMWE15 to RAMWE0 bits corresponding to the page to be accessed are set to 1, writing to the on-chip RAM becomes enabled; while if the RAMWE bits are cleared to 0, the on-chip RAM cannot be written to. In the access disabled state, a write to the page is ignored. The initial values of the RAMWE bits are 1.

To rewrite the RAMWE15 to RAMWE0 bits in RAMWEN0, word size data with upper byte as H'69 and lower byte as write data must be written. Similarly, to rewrite the RAMWE15 to RAMWE0 bits in RAMWEN1, word size data with upper byte as H'68 and lower byte as write data must be written.

:

To rewrite the RAMWE15 to RAMWE0 bits in RAMWEN0, execute an instruction to read from RAMWEN0 together with five or more NOP instructions immediately after an instruction to write to RAMWEN0. To rewrite the RAMWE15 to RAMWE0 bits in RAMWEN1, execute instructions for RAMWEN1 in the same way as for RAMWEN0.

28.2.6 RAM Access Cycle Set Register (RAMACYC) Amended

1422

... and ECC enable/disable. For details, refer to the recommended setting range shown in table 28.4.

| Item   | Page  | Revision (See Manual for Details)  |  |  |  |  |  |
|--|-------|--|--|--|--|--|--|
| 28.2.6 RAM Access                                    | 1423, | Amended  |  |  |  |  |  |
| Cycle Set Register<br>(RAMACYC)                      | 1424  | Accesses through the I bus are ruled by the peripheral clock $(P\phi)$ ; this is not the case with M or F buses, which are ruled by the internal clock $(\phi)$ . Therefore the number of cycles required for an I-bus access is determined by combination of the set PLL and peripheral clock multiplication ratios as well as the set value in this register.  |  |  |  |  |  |
|  |       | It takes one cycle to receive a command or address during I bus read access on condition that the PLL multiplication ratio is $\times 4$ or $\times 6$ or on condition that the PLL multiplication ratio is $\times 8$ or $\times 10$ and the peripheral clock multiplication ratio is $\times 2$ . It also takes one cycle during I bus write access on condition that the PLL multiplication ratio is $\times 4$ or $\times 6$ and the peripheral clock multiplication ratio is $\times 2$ . This is why the entire process, receiving a command and address, needs two cycles on such conditions. |  |  |  |  |  |
| Table 28.4   | 1424  | Terms amended  |  |  |  |  |  |
| Recommended<br>Settings for Read<br>and Write Cycles |       | [Before amendment] Rate $\rightarrow$ [After amendment] Ratio [Before amendment] PCLK $\rightarrow$ [After amendment] Peripheral Clock   |  |  |  |  |  |
| 28.3 On-Chip RAM                                     | 1426  | Amended  |  |  |  |  |  |
| Operations   |       | Accessing each area of the on-chip RAM is enabled or disabled by the RAME15 to RAME0 bits in the RAM enable control registers 0 and 1 (RAMEN0 and RAMEN1). When the RAME15 to RAME0 bits of RAMEN0 and RAMEN1 are cleared to 0, the on-chip RAM cannot be accessed. In this case, values read from the on-chip RAM are undefined and the on-chip RAM cannot be modified.   |  |  |  |  |  |
|  |       | Writing to each area of the on-chip RAM is enabled or disabled by the RAMWE15 to RAMWE0 bits in the RAM write enable control registers 0 and 1 (RAMWEN0 and RAMWEN1).  |  |  |  |  |  |
|  |       | Whenever the RAM error status register is set, a RAM error (RAME) interrupt can be generated. The interrupt generation can be set enabled or disabled via the RAM error interrupt control register (RAMINT).   |  |  |  |  |  |
| 28.5.2 State After                                   | 1429  | Deleted  |  |  |  |  |  |
| Turning on Power                                     |       | To enable correct (initialize) correspondence among RAM data, error correction data, and parity after power-on, write <del>longword (32 bits)</del> data to all on-chip RAM areas. If RAM is read without initialization, a RAM error may occur. Note that no RAM error occurs in RAM writes.  |  |  |  |  |  |

| Item                                  | Page       | Revision (See Manual for Details)   |   |            |            |                     |            |            |            |           |           |
|---------------------------------------|------------|---|---|------------|------------|---------------------|------------|------------|------------|-----------|-----------|
| 31.1 Register                         | 1446       | TEMS  | K0 to T   | EMSK       | 7, Regi    | ster Na             | me am      | ended      |            |           |           |
| Addresses (grouped by module name, in | to<br>1449 | DMA TE flag mask setting register 0   |   |            |            |                     |            |            |            |           |           |
| ordered of the                        | 1449       | DMA TE flag mask setting register 1   |   |            |            |                     |            |            |            |           |           |
| corresponding                         |            | DMA 7   | DMA TE flag mask setting register 2   |            |            |                     |            |            |            |           |           |
| section numbers)                      |            | DMA 7   | E flag  | mask s     | setting    | register            | 3          |            |            |           |           |
|                                       |            | DMA 7   | E flag  | mask s     | setting    | register            | 4          |            |            |           |           |
|                                       |            | DMA 7   | E flag  | mask s     | setting    | register            | 5          |            |            |           |           |
|                                       |            | DMA 7   | E flag  | mask s     | setting    | register            | 6          |            |            |           |           |
|                                       |            | DMA 7   | E flag  | mask s     | setting    | register            | 7          |            |            |           |           |
| 31.1 Register                         | 1450       | DMAF  | R, Reg  | ister N    | ame ar     | nended              |            |            |            |           |           |
| Addresses (grouped by module name, in |            | DMA c   | DMA operation flag bit register   |            |            |                     |            |            |            |           |           |
| ordered of the                        |            | ADMAABR, Register Name amended  |   |            |            |                     |            |            |            |           |           |
| corresponding                         |            | A-DMAC alias base register  |   |            |            |                     |            |            |            |           |           |
| section numbers)                      | 1480<br>to | Access Sizes of registers SSLPA/B/C, SPSRA/B/C, SPSSRA/B/C and SSLNDA/B/C amended |   |            |            |                     |            |            |            |           |           |
|                                       | 1482       | [Before amendment] 8, 16 $\rightarrow$ [After amendment] 8                        |   |            |            |                     |            |            |            |           |           |
|                                       | 1525       |   | Access Sizes of registers RAMEN0, RAMWEN0, RAMEN1, RAMWEN1 and RAMECC amended |            |            |                     |            |            |            |           |           |
|                                       |            | [Before   | e amer  | dment      | ] 8, 16    | $\rightarrow$ [Afte | r amen     | dment]     | 8, (16)    | )         |           |
| 31.2 List of Register                 | 1540       | Amend   | ded   |            |            |                     |            |            |            |           |           |
| Bits                                  |            | Module  | Register<br>Name  | 31/23/15/7 | 30/22/14/6 | 29/21/13/5          | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|                                       |            | A-DMAC  | ADMAIE0   |            |            |                     |            |            |            | =         | =         |
|                                       |            |   | ADMAIE9   | =          | =          | =                   | =          | =          |            | =         | =         |
|                                       |            |   | ADMADV0   |            |            |                     |            |            |            | _         | _         |
|                                       |            |   | ADMATE2   |            | =          | =                   | =          |            |            | =         |           |

31.2 List of Register 1541 Bits

Amended

| Module | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|--------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| A-DMAC | ADMADE0          |            |            |            |            |            |            | =         | =         |
|        | ADMADE7          | _          | -          | =          | =          |            |            | =         |           |
|        | ADMAMO<br>DE0    |            |            |            |            | =          | =          | =         | -         |

1623 Amended

| Module   | Register<br>Name | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
|----------|------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| I/O port | PDPR             | =          | =          | PD13PR     | PD12PR     | PD11PR     | PD10PR     | PD9PR     | PD8PR     |
|          |                  | PD7PR      | PD6PR      | PD5PR      | PD4PR      | PD3PR      | PD2PR      | PD1PR     | PD0PR     |

1628 Amended

|        | Register |            |            |            |            |            |            |           |           |
|--------|----------|------------|------------|------------|------------|------------|------------|-----------|-----------|
| Module | Name     | 31/23/15/7 | 30/22/14/6 | 29/21/13/5 | 28/20/12/4 | 27/19/11/3 | 26/18/10/2 | 25/17/9/1 | 24/16/8/0 |
| RAM    | RAMINT   | _          | _          | RPEIE1     | RPEIE0     | REDIE1     | RECIE1     | REDIE0    | RECIE0    |

31.3 Register States 1703 in Each Operating Mode

Amended

|        |          | Reset State | Power-Down State |          |  |  |  |
|--------|----------|-------------|------------------|----------|--|--|--|
|        | Register |             | Hardware         |          |  |  |  |
| Module | Name     | Power-On    | Standby          | Sleep    |  |  |  |
| JTAG   | SDIR     | Undefined   | Undefined        | Retained |  |  |  |

1704 Amended

|        |                  | Reset State | Powe                | r-Down State |
|--------|------------------|-------------|---------------------|--------------|
| Module | Register<br>Name | Power-On    | Hardware<br>Standby | Sleep        |
| AUD-II | AUWASR           | Undefined   | Undefined           | Retained     |
|        | AUWAER           | Undefined   | Undefined           | Retained     |
|        | AUWBSR           | Undefined   | Undefined           | Retained     |
|        | AUWBER           | Undefined   | Undefined           | Retained     |
|        |                  |             |                     |              |

| Item                                  | Page | Revision (  | See Manual f         | or Deta            | ails)           |             |                  |                |  |  |
|---------------------------------------|------|---|----------------------|--------------------|-----------------|-------------|------------------|----------------|--|--|
| Table 32.2                            | 1712 | Deleted   | Deleted              |                    |                 |             |                  |                |  |  |
| Correspondence between Power          |      | Pin No.   | Symbol               | Permi              | ssible Inpu     | t Voltage   | (V)              |                |  |  |
| Supply Names and<br>Pins              |      | B11   | WDTOVF               | ¥ <sub>∞</sub> + € | <del>).3</del>  |             |                  |                |  |  |
| Table 32.4 DC                         | 1723 | Amended   |                      |                    |                 |             |                  |                |  |  |
| Characteristics Input                 |      | Item  |                      |                    | Symbol          | Min.        | Тур.             | Max.           |  |  |
| Level Voltage                         |      | TTL input volt  | -                    | to PC0             | V <sub>IH</sub> | PVcc1 × 0.7 | _                | PVcc1<br>+ 0.3 |  |  |
|                                       |      |   |                      |                    | V <sub>IL</sub> | -0.3        | _                | PVcc1<br>× 0.3 |  |  |
| Table 32.5 DC                         | 1725 | Item amend  | led                  |                    |                 |             |                  |                |  |  |
| Characteristics Input<br>Leak Current |      | [Before ame   | endment] FEV         | V → [A             | fter amer       | ndment]     | FWE              |                |  |  |
| Table 32.12 DC                        | 1732 | Amended   |                      |                    |                 |             |                  |                |  |  |
| Characteristics —<br>Standby          |      | Item  |                      | Meas               | surement C      | Condition   | S                |                |  |  |
|                                       |      | Supply curren<br>(V <sub>cc</sub> power su                | •                    | -40°C ≤ Ta ≤ 50°C  |                 |             |                  |                |  |  |
|                                       |      |   |                      | 50°C               | < Ta ≤ 105      | 5°C         |                  |                |  |  |
|                                       |      |   |                      | 105°               | C < Ta ≤ 12     | 25°C        |                  | <u> </u>       |  |  |
|                                       |      |   |                      |                    |                 |             |                  |                |  |  |
| Table 32.21 Serial Communications     | 1751 | Amended   |                      |                    |                 |             |                  |                |  |  |
| Interface Timing                      |      | Item  |                      |                    |                 |             | Symb             | ool            |  |  |
| -                                     |      | Synchronous<br>(SCK is inpu                               | s mode receive<br>t) | data se            | tup time        |             | t <sub>RxS</sub> |                |  |  |
|                                       |      | Synchronous<br>(SCK is input                              | s mode receive<br>t) | data ho            | old time        |             | t <sub>RxH</sub> |                |  |  |
|                                       |      | Synchronous mode transmit data delay time (SCK is output) |                      |                    |                 |             |                  |                |  |  |
|                                       |      | (SCK is outp  | out)                 |                    |                 |             |                  |                |  |  |

## Index

| Numerics                              | В   |
|---------------------------------------|---|
| 16-bit/32-bit displacement            | Banked registers and input/output         |
| •                                     | method                                    |
|                                       | Bathtub curve                             |
| A                                     | Bit manipulation instructions79           |
| A/D converter (ADC)                   | Block diagram of subblocks of timer F 546 |
| Absolute accuracy                     | Block diagram of timer A401               |
| Absolute address                      | Block diagram of timer B423               |
| Absolute address accessing            | Block diagram of timer C460               |
| Absolute maximum ratings              | Block diagram of timer D485               |
| Access size and data alignment        | Block diagram of timer E526               |
| Access wait control                   | Block diagram of timer G589               |
| Address array                         | Block diagram of timer H597               |
| Address errors 109                    | Block diagram of timer J605               |
| Address map                           | Boot mode1253, 1353                       |
| Addressing modes48                    | Boundary scan commands995                 |
| A-DMAC interface                      | Branch instructions73                     |
| Advanced timer unit III (ATU-III) 361 | Branch source address1037                 |
| Advanced user debugger II             | Branch tracing1036                        |
| (AUD-II)1017                          | Break detection and processing738         |
| Alias areas                           | Break on data access cycle202             |
| Allowable settings for pins           | Break on instruction fetch cycle201       |
| AVrefh_A/AVrefh_B and                 | Burst mode                                |
| AVrefl_A/AVrefl_B986                  | Bus connections in RAM1390                |
| Analog input sampling and             | Bus Operation in data transfer by         |
| scan conversion time                  | A-DMAC352                                 |
| Analog input voltage range 986        | Bus state controller (BSC)209             |
| Arithmetic operation instructions 68  | BYPASS command1013                        |
| Asynchronous mode 691, 718            |   |
| AUD bus command                       |   |
| AUD tracing mode1023                  | C   |
| Automotive direct memory              | Cache configuration                       |
| access controller (A-DMAC)309         | Calculating exception handling            |
| Auto-request mode                     | vector table addresses106                 |
| -                                     | CAN bus interface919                      |
|                                       | CAN interface818                          |
|                                       | Channel priorities357                     |
|                                       | _   |

| Channel priority                        | DC characteristics —              |      |
|---|-----------------------------------|------|
| CLAMP command                           | injection current values          | 1729 |
| Clock pulse generator (CPG)85           | DC characteristics —              |      |
| Clock source                            | input capacitance                 | 1730 |
| Clock synchronous mode 691, 727         | DC characteristics —              |      |
| CMCNT count timing 685                  | output level voltage              | 1727 |
| Combination of TSEG1 and TSEG2 850      | DC characteristics —              |      |
| Common controller registers 365         | permissible output current values | 1729 |
| Compare match function 480              | DC characteristics — standby      | 1732 |
| Compare match timer (CMT) 679           | DC characteristics                |      |
| Conditions for determining number       | input leak current                | 1725 |
| of idle cycles243                       | DC characteristics                |      |
| Configuration of RCAN-TL1 893           | input level voltage               | 1723 |
| Conflict between word-write and         | DC characteristics                |      |
| count-up processes of CMCNT 689         | pull-up/pull-down MOS current     | 1726 |
| Conflict between write and              | Definition of A/D conversion      |      |
| compare-match processes of CMCNT 688    | accuracy                          | 984  |
| Connecting crystal resonator 88         | Delayed branch instructions       | 45   |
| Continuous scan conversion mode 972     | Descriptions of timer A registers | 402  |
| Controller area network (RCAN-TL1) 813  | Descriptions of timer C           | 461  |
| Controlling RSPI pins775                | Descriptions of timer D           | 486  |
| Correspondence between                  | Descriptions of timer E           | 527  |
| power supply names and pins 1712        | Descriptions of timer F           | 548  |
| CPU33                                   | Descriptions of timer G           | 590  |
| CSn assert period expansion             | Descriptions of timer H           | 598  |
| Cycle stealing mode298                  | Descriptions of timer J           | 606  |
|   | Direct memory access controller   |      |
|   | (DMAC)                            | 251  |
| D                                       | Displacement accessing            | 48   |
| Data array                              | DMA transfer requests             |      |
| Data format                             | DMAC interface                    | 918  |
| Data format in registers                | Dual address mode                 | 297  |
| Data formats in memory                  |                                   |      |
| Data retention at hardware standby 1427 |                                   |      |
| Data retention at reset 1427            | ${f E}$                           |      |
| Data transfer instructions              | Edge counting                     | 574  |
| Data transfer with                      | Edge interval measuring block     | 421  |
| interrupt request signals               | Edge interval measuring function  |      |
| DC characteristics —                    | and edge input stopping function  | 447  |
| current consumption 1731                | EEPROM                            | 1329 |

| Effective address calculation            | General registers                   | 33       |
|--|-------------------------------------|----------|
| Electrical characteristics               | Global base register (GBR)          | 36       |
| Endian                                   |                                     |          |
| Error protection                         |                                     |          |
| Error protection types                   | Н                                   |          |
| Error Protection Types1370               | Halt mode                           | 894      |
| Example of address comparison 1042       | Hardware prefetch                   |          |
| Example of time triggered system 908     | Hardware protection13               |          |
| Exception handling 101                   | Hardware standby mode               |          |
| Exception handling state                 | HIGHZ command                       |          |
| Exception handling vector table 105      |                                     |          |
| Exception source generation              |                                     |          |
| immediately after                        | ī                                   |          |
| delayed branch instruction119            | I/O ports                           | 1145     |
| Exceptions triggered by instructions 115 | ID reorder                          |          |
| External clock input                     | IDCODE command                      |          |
| External space interface                 | Immediate data                      |          |
| EXTEST command                           | Immediate data accessing            |          |
|  | Immediate data format               |          |
|  | In minimum time-at-level            |          |
| F  | cancellation mode                   | 412, 476 |
| FCU command list1288                     | Initial values of control registers |          |
| FCU command usage 1296, 1364             | Initial values of floating-point    |          |
| Fixed mode                               | registers                           | 41       |
| Floating-point operation instructions 77 | Initial values of floating-point    |          |
| Floating-point registers37               | system registers                    | 41       |
| Floating-point system registers38        | Initial values of general registers |          |
| Flowchart of DMA transfer 344            | Initial values of                   |          |
| Four-time multiplication event count 586 | port A control registers            | 1063     |
| FPU-related CPU instructions             | Initial values of                   |          |
| Frequency ranges and clock selection 87  | port B control registers            | 1077     |
| Frequency-multiplied clock generator 449 | Initial values of                   |          |
| Frequency-multiplied                     | port C control registers            | 1092     |
| clock signal corrector                   | Initial values of system registers  |          |
| Full scale error                         | Initializing RSPI                   |          |
| Full tracing mode                        | Input capture                       |          |
|  | Input capture function              |          |
|  | Instruction features                |          |
| G  | Instruction format                  | 53       |
| General illegal instructions 117         | Instruction set                     |          |

| Integer division exceptions                | M                                     |
|--|---------------------------------------|
| Integer to floating-point conversion       | Mailbox817, 821                       |
| operation                                  | Mailbox control817                    |
| Interface between CPU and                  | Master mode operation798              |
| A/D converter (ADC)                        | Measurement of                        |
| Internal arbitration for transmission 898  | PWM input waveform timing579          |
| Internal state after reset cancellation 99 | Measurement of time during            |
| Interrupt controller (INTC)                | high/low input levels577              |
| Interrupt conversion                       | Memory error interrupt 139            |
| during scan conversion                     | Message control field827              |
| Interrupt exception handling 114           | Message data fields832                |
| Interrupt exception handling vectors       | Message receive sequence912           |
| and priority levels141                     | Message transmission request 897, 907 |
| Interrupt priority level113                | Micro processor interface (MPI)817    |
| Interrupt requests on completion of        | Minumum time-at-level                 |
| interrupt conversion                       | cancellation mode620                  |
| Interrupt requests on completion of        | Module standby function1439           |
| scan conversion                            | Monitoring via ADEND_A and            |
| Interrupt response time                    | ADEND_B output pins982                |
| Interrupts                                 | MOSI signal value determination       |
| Interval count operation                   | during SSL negate period776           |
| IRQ interrupts                             | Multi-input signature generator       |
|  | (MISG)1213                            |
|  | Multiple receive error occurrence737  |
| ${f J}$                                    | Multiplex pin list 1051, 1052, 1053.  |
| JTAG interface991                          |                                       |
| Jump table base register (TBR) 36          |                                       |
|  | Multiplied clock generating block 421 |
|  | Multiply and accumulate register high |
| L  | (MACH)37                              |
| List of AUD bus commands 1033              | Multiply and accumulate register low  |
| List of functions assigned to              | (MACL)37                              |
| each channel                               | Multiply/multiply-and-accumulate      |
| Load-store architecture                    | operations45                          |
| Local acceptance filter mask (LAFM) 831    |                                       |
| Logic operation instructions               |                                       |
| Loopback mode811                           | N                                     |
| 200poues 1110 <b>40</b>                    | NMI interrupt138                      |
|  | Nonlinearity error984                 |
|  | Note on board design90, 987           |
|  | 2                                     |

| Note on connecting power supply           | Overview of timer G           | 589     |
|---|-------------------------------|---------|
| for PLL oscillator91                      | Overview of timer H           | 597     |
| 0   | Overview of timer J           | 605     |
| Offset error                              |                               |         |
| On-chip peripheral module interrupts 139  |                               |         |
| On-chip peripheral module request 288     | P                             |         |
| On-chip RAM address space 1391            | Package Dimensions            | 1785    |
| On-chip RAM data retention                | Page conflict                 |         |
| during reset                              | Pin descriptions              |         |
| Operating modes83                         | in AUD tracing mode           | 1021    |
| Operating modes and data bus width        | Pin descriptions              |         |
| in CS0 space                              | in RAM monitoring mode        | 1022    |
| Operation example of                      | Pin function controller (PFC) |         |
| one-shot pulse output 520, 522, 523       | Pin States                    | 1777    |
| Operation for A-DMAC channels             | Port A                        | 1148    |
| used for ATU-III (timers A, C, and F) 338 | Port B                        | 1152    |
| Operation for A-DMAC channels             | Port C                        | 1159    |
| used for RCAN345                          | Port D                        | 1163    |
| Operation for A-DMAC channels             | Port E                        | 1167    |
| used for RSPI and SCI340                  | Port F                        | 1172    |
| Operation of free-running counter 418     | Port G                        | 1179    |
| Operation of noise canceler 415           | Port H                        | 1187    |
| Operation of PWM 542, 543, 544            | Port J                        | 1190    |
| Operations of timer A415                  | Port K                        | 1198    |
| Operations of timer B447                  | Port L                        | 1205    |
| Operations of timer C479                  | Port output disable (POD)     | 1210    |
| Operations of timer D 518                 | Power-down modes              | 1431    |
| Operations of timer E541                  | Power-down state              | 82      |
| Operations of timer F 574                 | Power-on reset                | 107     |
| Operations of timer G596                  | Preceding edge                |         |
| Operations of timer H 604                 | cancel mode 569, 570, 5       | 72, 573 |
| Operations of timer J 622                 | Premature-transition          |         |
| Overview of common controller 387         | cancellation mode412, 4       | 76, 620 |
| Overview of prescalers 397                | Prescaler registers           | 365     |
| Overview of timer A                       | Procedure register (PR)       | 37      |
| Overview of timer B                       | Product Code Lineup           | 1784    |
| Overview of timer C                       | Product Information MAT       | 1371    |
| Overview of timer D                       | Program counter (PC)          | 37      |
| Overview of timer E 525                   | Program execution state       | 82      |
| Overview of timer F545                    |                               |         |

| Programming/erasing                       | ADANS3             | 948          |
|---|--------------------|--------------|
| host command wait state 1276              | ADCER0 and ADCER1  | 943          |
| Protection                                | ADCSR0 and ADCSR 1 | 940          |
| PV <sub>CC</sub> 1 voltage in             | ADMAABR            | 322          |
| each operating mode                       | ADMAAR             | 329          |
| PWM function 482                          | ADMABUF            | 331          |
|   | ADMADE             | 326          |
|   | ADMADV             | 324          |
| Q   | ADMAIE             | 323          |
| Quantization error984                     | ADMAMODE           | 327          |
|   | ADMAOR             | 321          |
|   | ADMARAR            | 330          |
| R   | ADMARTCR           | 330          |
| RAM1389                                   | ADMARVPR           | 332          |
| RAM block diagram                         | ADMATCR            | 328          |
| RAM data retention                        | ADMATE             | 325          |
| RAM monitoring mode                       | ADMATVPR           | 333          |
| RCAN-TL1 control registers                | ADR0 to ADR27,     |              |
| RCAN-TL1 interrupt sources                | ADR40 to ADR48,    |              |
| RCAN-TL1 mailbox registers                | ADRD0, and ADRD1   | 935          |
| RCAN-TL1 memory map                       | ADREF0 and ADREF1  | 949          |
| RCAN-TL1 timer registers                  | ADSTRG0            | 960          |
| Realtime tracing mode                     | ADSTRG1            | 961          |
| Receive data sampling timing and          | ADTRD0             | 965          |
| receive margin (asynchronous mode) 738    | ADTRD1             | 967          |
| Reconfiguration of mailbox                | ADTRE0             | 955          |
| Register bank                             | ADTRE1             | 956          |
| error exception handling                  | ADTRF0             | 962          |
| Register bank errors                      | ADTRF1             | 964          |
| Register bank exception                   | ADTRS0             | 957          |
| Register banks                            | ADTRS1             | 959          |
| Register descriptions of timer B          | ARCR4 to ARCR7     |              |
| Registers                                 | ATUENR             | 388          |
| ABACK0870                                 | AUCSR              | 1023         |
| ABACK1870                                 | AUECSR             | 1029         |
| ADADC0 and ADADC1                         | AUWAER             | 1027         |
| ADADS0951                                 | AUWASR             | 1027         |
| ADADS1 952                                | AUWBER             | 1028         |
| ADANS0                                    | AUWBSR             |              |
| ADANS1                                    | BAMR180, 18        | 34, 188, 192 |
| 7.10.10.10.10.10.10.10.10.10.10.10.10.10. |                    |              |

| BAR 179, 183, 187, 191           | EEPWE0            | 1344       |
|----------------------------------|-------------------|------------|
| BBR 181, 185, 189, 193           | EEPWE1            | 1345       |
| BCR0849                          | FAEINT            | 1228, 1340 |
| BCR1 847                         | FASTAT            | 1225, 1337 |
| BRCR 195                         | FCMDR             | 1248       |
| CBCNT391                         | FCPSR             | 1249       |
| CCR 885                          | FCRJ0 and FCRJ1   | 610        |
| CDRF0 to CDRF19567               | FCURAME           | 1231       |
| CHCR0 to CHCR7261                | FDNRJ0 and FDNRJ1 | 619        |
| CHFR0 to CHFR7267                | FENTRYR           |            |
| CKCR 1209                        | FIFOJ0 and FIFOJ1 | 618        |
| CMAX_TEW 880                     | FMODR             | 1224, 1336 |
| CMCNT684                         | FPESTAT           | 1250       |
| CMCOR684                         | FPMON             | 1223       |
| CMCR682                          | FPROTR            |            |
| CMPOD0 and CMPOD1509             | FPSCR             |            |
| CMSR 683                         | FPUL              |            |
| CMSTR681                         | FRAMECCR          | 1238       |
| CRLDE00 to CRLDE53 539           | FRESETR           |            |
| CSnBCR ( $n = 0 \text{ to } 3$ ) | FSTATR0           | 1232       |
| CSnWCR ( $n = 0 \text{ to } 3$ ) | FSTATR1           |            |
| CYCTR886                         | GRAF0 to GRAF19   |            |
| CYLRE00 to CYLRE53538            | GRBF0 to GRBF19   |            |
| DAR0 to DAR7                     | GRC00 to GRC43    |            |
| DCNTD00 to DCNTD 33516           | GRCF0 to GRCF19   |            |
| DCRD0 to DCRD3499                | GRD00 to GRD33    |            |
| DMAFR                            | GRDF12 to GRDF15  | 566        |
| DMAOR277                         | GSR               | 844        |
| DMARS0 to DMARS3282              | IBCR              | 134        |
| DMATCR0 to DMATCR7261            | IBNR              | 135        |
| DRLDE00 to DRLDE53540            | ICR0              |            |
| DSRD0 to DSRD3497                | ICR1              |            |
| DSTRD0 to DSTRD3496              | ICRA0 to ICRA5    |            |
| DTRE00 to DTRE53539              | ICRB0             |            |
| ECNTAF0 to ECNTAF19 559          | ICRB1             |            |
| ECNTBF0 to ECNTBF19560           | ICRB2             |            |
| ECNTCF0 to ECNTCF19562           | IMR               |            |
| EEPMAT 1350                      | IPR01 to IPR29    |            |
| EEPRE0                           | IRQRR             |            |
| EEPRE11343                       | IRR               | 852        |

| LDB436                     | PBDSR | 1157 |
|----------------------------|-------|------|
| MBIMR0 875                 | PBIOR | 1076 |
| MBIMR1 874                 | PBIR  | 1155 |
| MCR 839                    | PBPR  | 1154 |
| MISR 1215                  | PBPSR | 1158 |
| MISRCDR 1214               | PCCR1 | 1104 |
| MISRCR                     | PCCR2 | 1101 |
| NCCRC0 to NCCRC4462        | PCCR3 | 1097 |
| NCCRF550                   | PCCR4 | 1093 |
| NCMR 393                   | PCDR  | 1159 |
| NCNTA0 to NCNTA5412        | PCIOR | 1091 |
| NCNTC00 to NCNTC43 476     | PCPR  | 1161 |
| NCNTFA0 to NCNTFA19 568    | PDCR1 | 1110 |
| NCNTFB0 to NCNTFB2570      | PDCR2 | 1108 |
| NCNTJ0 and NCNTJ1 620, 621 | PDDR  | 1163 |
| NCRA0 to NCRA5414          | PDIOR | 1107 |
| NCRFA0 to NCRFA19 572      | PDIR  | 1166 |
| NCRFB0 to NCRFB2573        | PDPR  | 1165 |
| NCRJ0 and NCRJ1 621        | PECR1 | 1115 |
| OCR1H602                   | PECR2 | 1113 |
| OCRB0 432                  | PEDR  | 1167 |
| OCRB1433                   | PEDSR | 1171 |
| OCRB6441                   | PEIOR | 1112 |
| OCRB7 442                  | PEIR  | 1170 |
| OCRD00 to OCRD33514        | PEPR  | 1169 |
| OCRG0 to OCRG5595          | PFCR1 | 1120 |
| OCRJ0 and OCRJ1 617        | PFCR2 | 1118 |
| OSBRD0 to OSBRD3511        | PFDR  | 1172 |
| PACR11073                  | PFDSR |      |
| PACR21070                  | PFIOR | 1117 |
| PACR3                      | PFIR  | 1176 |
| PACR41064                  | PFPR  | 1175 |
| PADR 1148                  | PFPSR |      |
| PAIOR 1062                 | PGCR1 |      |
| PAPR1150                   | PGCR2 |      |
| PBCR11087                  | PGDR  |      |
| PBCR2                      | PGDSR |      |
| PBCR3                      | PGER  |      |
| PBCR41077                  | PGIOR |      |
| PBDR1152                   | PGIR  | 1183 |

| PGPR 1182                | RDMATCR0 to RDMATCR7 | 274  |
|--------------------------|----------------------|------|
| PHCR1129                 | REC                  | 862  |
| PHDR1187                 | RFMK                 | 887  |
| PHIOR1128                | RFPR0                | 873  |
| PHPR1189                 | RFPR1                | 873  |
| PIMR439                  | RFTROFF              | 882  |
| PJCR11133                | RLDB                 | 437  |
| PJCR21132                | RLDCRE0 to RLDCRE5   | 532  |
| PJDR1190                 | ROMMAT               | 1230 |
| PJDSR1194                | RSAR0 to RSAR7       | 272  |
| PJIOR1131                | RXPR0                | 872  |
| PJIR1193                 | RXPR1                | 871  |
| PJPR1192                 | SAR0 to SAR7         | 260  |
| PJPSR1196                | SCBRR1               | 710  |
| PKCR11138                | SCRDR1               | 696  |
| PKCR21136                | SCRSR1               | 695  |
| PKDR1198                 | SCSCR1               | 700  |
| PKDSR1202                | SCSMR1               | 697  |
| PKIOR1135                | SCSSR1               | 703  |
| PKIR1201                 | SCTDR1               | 697  |
| PKPR1200                 | SCTSR1               | 696  |
| PKPSR1203                | SDBPR                | 995  |
| PLCR11142                | SDBSR                | 996  |
| PLCR2 1141               | SDID                 | 995  |
| PLDR1205                 | SDIR                 | 995  |
| PLIOR1140                | SINTR1 to SINTR15    | 137  |
| PLIR1208                 | SPBR                 | 763  |
| PLPR1207                 | SPCKD                | 764  |
| PSCR0 to PSCR3398        | SPCMD                | 768  |
| PSCRE0 to PSCRE5530      | SPCR                 | 751  |
| RAMACYC1422              | SPDR                 | 759  |
| RAMECC 1414              | SPND                 | 767  |
| RAMEN0 and RAMEN11393    | SPPCR                | 754  |
| RAMERR 1415              | SPSCR                | 760  |
| RAMINT 1419              | SPSR                 | 756  |
| RAMWEN0 and RAMWEN1 1404 | SPSSR                | 761  |
| RARCR4 to RARCR7276      | SSLND                | 766  |
| RCCR1378                 | SSLP                 | 753  |
| RCCR2 1380               | SSTRE0 to SSTRE5     | 528  |
| RDAR0 to RDAR7273        | STBCR                | 1433 |

| TCCLRB44               | 16    | TIOR2D0 to TIOR2D3     | 493 |
|------------------------|-------|------------------------|-----|
| TCMR0, TCMR1, TCMR2 88 | 37    | TIORB                  | 425 |
| TCNT1D0 to TCNT1D351   | 12    | TIORC0 to TIORC4       | 471 |
| TCNT1H60               | )1    | TOCRD0 to TOCRD3       | 510 |
| TCNT2D0 to TCNT2D351   | 13    | TOCRE0 to TOCRE5       | 536 |
| TCNT2H60               | )3    | TSR                    | 883 |
| TCNTA41                | 11    | TSRA                   | 407 |
| TCNTB043               | 31    | TSRB                   | 427 |
| TCNTB143               | 33    | TSRC0 to TSRC4         | 467 |
| TCNTB243               | 38    | TSRD0 to TSRD3         | 501 |
| TCNTB344               | 13    | TSRE0 to TSRE5         | 532 |
| TCNTB444               | 14    | TSRF0 to TSRF19        | 556 |
| TCNTB544               | 15    | TSRG0 to TSRG5         | 592 |
| TCNTB644               | 10    | TSRH                   | 599 |
| TCNTC0 to TCNTC447     | 74    | TSRJ0 and TSRJ1        | 612 |
| TCNTE00 to TCNTE53 53  | 37    | TSTRC                  | 461 |
| TCNTG0 to TCNTG5 59    | 94    | TSTRD                  | 486 |
| TCNTJ0 and TCNTJ161    | 17    | TSTRE                  | 527 |
| TCNTR88                | 36    | TSTRF                  | 548 |
| TCRA 40                | )2    | TSTRG                  | 590 |
| TCRB42                 | 24    | TSTRJ                  | 606 |
| TCRC0 to TCRC4 46      | 54    | TTCR0                  | 877 |
| TCRD0 to TCRD348       | 37    | TTTSEL                 | 889 |
| TCRE0 to TCRE5 53      | 31    | TXACK0                 | 869 |
| TCRF0 to TCRF19 55     | 52    | TXACK1                 | 868 |
| TCRG0 to TCRG559       | 91    | TXCR0                  | 868 |
| TCRH 59                | 98    | TXCR1                  | 867 |
| TCRJ0 and TCRJ160      | )7    | TXPR0                  | 866 |
| TEC 86                 | 52    | TXPR1                  | 865 |
| TEMSK0 to TEMSK7 27    | 70    | UMSR0                  | 876 |
| TIERA 40               | )9    | UMSR1                  | 875 |
| TIERB 42               | 29    | WRCR                   | 671 |
| TIERC0 to TIERC447     | 70    | WTCNT                  | 668 |
| TIERD0 to TIERD3 50    | )7    | WTCR                   | 665 |
| TIERE0 to TIERE5 53    | 35    | WTSR                   | 669 |
| TIERF0 to TIERF1955    | 55 Re | elationship between    |     |
| TIERJ0 and TIERJ161    | 16 RS | SPI modes and SPCR and |     |
| TIOR1A40               | )4 de | scription of each mode | 773 |
| TIOR1D0 to TIOR1D349   |       | eliability1            |     |
| TIOR2A40               |       |                        |     |

| Renesas seriai peripherai interface      | Slot illegal exceptions            | 110      |
|--|------------------------------------|----------|
| (RSPI)743                                | Software interrupts (SINT)         | 139      |
| Reset at power-on98                      | Software protection13              | 19, 1368 |
| Reset during operation                   | SRAM access timing                 | 246      |
| Reset operation                          | SRAM interface with byte selection | 236      |
| Reset sequence                           | Stack after                        |          |
| Reset state                              | interrupt exception handling       | 161      |
| Restoring from bank                      | Stack status after                 |          |
| Restoring from stack                     | exception handling ends            | 120      |
| RISC-type instruction set44              | Starting prescaler                 | 399      |
| Roles of mailboxes 823                   | Starting scan conversion           |          |
| ROM1217                                  | with ATU-III timer trigger         | 981      |
| ROM cache (ROMC)1375                     | Starting scan conversion           |          |
| Rotation speed/pulse measurement 581     | with external trigger              | 980      |
| Round-robin mode293                      | State transition in boot mode      | 1254     |
| RSPI data format785                      | State transition of TAP controller | 1012     |
| RSPI error detection function            | Status register (SR)               | 35       |
| RSPI system configuration example 776    | Stopping prescaler                 | 399      |
| RSPI transfer format                     | Subsequent edge                    |          |
|  | cancel mode 568, 570,              | 572, 573 |
|  | Supported DMA transfers            | 296      |
| S  | Suspending operation               | 1312     |
| SAMPLE/PRELOAD command 1013              | System control instructions        | 75       |
| Saving to bank 168                       | System matrix                      | 838      |
| Saving to stack                          |                                    |          |
| SCI interrupt sources and A-DMAC 736     |                                    |          |
| SCTDR1 writing and TDRE flag 737         | Т                                  |          |
| Sending a break signal                   | T bit                              | 45       |
| Serial communication interface (SCI) 691 | TAP controller                     |          |
| Settable AUDCK clock ratios 1044         | Test mode settings                 |          |
| Setting I/O ports for RCAN-TL1920        | Time master (potential)            |          |
| SH72546R block diagram10                 | Time slave                         |          |
| SH72546R features                        | Time trigger control (TT control)  |          |
| SH72546R pin arrangements11              | Time trigger window                |          |
| SH72546R pin functions                   | Time triggered transmission        |          |
| Shift instructions72                     | Timer A registers                  |          |
| Sign extension of word data44            | Timer B registers                  |          |
| Single-cycle scan conversion mode 970    | Timer C registers                  |          |
| Slave mode operation 806                 | Timer D registers                  |          |
| Sleep mode                               | Timer E registers                  |          |

| Timer F registers                    |
|--------------------------------------|
| Timer G registers                    |
| Timer H registers                    |
| Timer J registers                    |
| Timestamp                            |
| Timing of compare match flag         |
| clearing                             |
| Timing of compare match flag         |
| setting 686                          |
| Timing to clear interrupt source 174 |
| Transfer enable and disable          |
| conditions for channels used for     |
| ATU-III (Timer G) and ADC            |
| Transfer enable and disable          |
| conditions for channels used for     |
| ATU-III (Timers A, C, and F)         |
| Transfer enable and disable          |
| conditions for channels used for     |
| RSPI and SCI359                      |
| Transmission buffer                  |
| empty/receive buffer full flags 790  |
| Trap instructions                    |
| Tx-trigger control field             |
| Tx-trigger time (TTT)                |
| Types of                             |
| exception handling and priority 101  |

## U

| U                                    |
|--------------------------------------|
| Unconditional branch instructions    |
| with no delay slot45                 |
| Up/down event count584               |
| User boot mode                       |
| User break controller (UBC)175       |
| User break interrupt                 |
| User program mode                    |
| Using WDT in interval timer mode 676 |
| Using WDT in watchdog timer mode 674 |
|                                      |
| $\mathbf{v}$                         |
| Valid edge interval counting576      |
| Value of saved program counter203    |
| Vector base register (VBR)36         |
|                                      |
| W                                    |
| Wait between access cycles242        |
| Watchdog timer (WDT)663              |
|                                      |

Window data tracing......1039

SH72546R User's Manual: Hardware

Publication Date: Rev.1.00 Mar 10, 2011

Rev.2.00 Apr 01, 2014

Published by: Renesas Electronics Corporation



## **SALES OFFICES**

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc. 2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A. Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited 1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Milliboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH Arcadiastrasse 10, 40472 Düsseldorf, Germany

Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong Tel: +852-2265-6688, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd. 80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949 Tel: +65-6213-0200, Fax:+65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd. 12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea Tel: +82-2-558-3737, Fax: +82-2-558-5141

SH72546R

